J2UL-1337-06Z0(00)

Red Hat Enterprise Linux 6
Linuxユーザーズマニュアル

FUJITSU

目次

本書をお読みになる前に 6
本書の構成 6
本書の表記 7
参照マニュアル 8
保守サービスについて 11
商標および著作権について 11
改版履歴 12

1 システム設計：ハードディスク **13**
1.1 ディスクパーティション 13
1.1.1 パーティションテーブル 13
1.1.2 ディスクパーティションの見積り方法 14
1.1.3 ディスクパーティションを作成するコマンド 16
1.1.4 partedコマンド使用時の注意事項 16
1.2 ファイルシステム 18
1.2.1 ファイルシステムの最大値 19
1.2.2 ファイルシステム使用上の注意事項 19
1.2.3 ファイルシステムのマウントオプション 20
1.3 ディスクの冗長化 21
1.3.1 LVM 21
1.3.2 MD 23
1.3.3 DM-MP 24
1.4 ディスク利用時の考慮点 27
1.4.1 450台を超えるディスク 27
1.4.2 ディスクパーティション作成時のデバイスファイル 28
1.4.3 ディスクの活性増設とUSBドライブ 28
1.4.4 rawデバイス 29

2 システム設計：ネットワーク **31**
2.1 LANの構成 31
2.1.1 LAN構成の考慮点 31
2.1.2 RHELシステムにおけるLAN接続機能 32
2.1.3 ネットワークデバイスの設定・管理 32
2.2 VLAN 35
2.2.1 VLANとは 35
2.2.2 VLANの設定 36
2.2.3 VLAN使用時の注意事項 38
2.3 bonding 40
2.3.1 動作モードと監視モード 40
2.3.2 2重化機能 42
2.3.3 ロードバランシング機能 44
2.3.4 bondingドライバとVLANの組合せ 45
2.3.5 コマンドによるbondingデバイスの操作 46

3 システム設計：デバイス名ずれを考慮した設計 **48**
3.1 デバイス名とデバイス名ずれ 48
3.2 デバイス名ずれによるトラブル 49
3.3 ディスク系デバイスにおける対処 50
3.3.1 udev機能の利用 50
3.3.2 ディスク交換のby-id名変更 52
3.3.3 ミドルウェアの利用 54
3.4 ネットワーク系デバイスにおける対処 54
3.4.1 udev機能の利用 54
3.4.2 ネットワーク系デバイスの交換・増設 60
3.4.3 レスキューモードでネットワーク系デバイスを使用する場合の注意 65

4 システム設計：ログファイル運用 **66**
4.1 ログファイルの運用設計 66
4.1.1 ログローテーション 66
4.1.2 ログファイルの種類 67
4.2 システムログファイルのログローテーション 69
4.2.1 ログローテーションとスケジュール 69
4.2.2 システムログファイルの切替え周期と保有世代数 69
4.2.3 システムログファイルのバックアップに関する注意事項 71
4.2.4 ログローテーションのスケジュール変更 71
4.2.5 ログローテーション条件の変更 72
4.3 ユーザー固有ログファイルのログローテーション 73
4.3.1 ログローテーションの検討事項 73
4.3.2 logrotateコマンドのログファイル切替え方式 73
4.3.3 システムログのログローテーションへの追加 76
4.3.4 logrotateコマンドを独自に使用する 77
4.4 ログローテーション設定ファイル 78
4.4.1 ログローテーション設定ファイルの種類 78
4.4.2 ログローテーション運用の設定ファイルの記述様式 79

5 システム設計：時刻同期（**NTP**） **82**
5.1 時刻の仕組みと時刻同期 82
5.1.1 時刻同期の必要性 82
5.1.2 ハードウェア時計とシステム時計 82
5.2 NTPの時刻同期方式 83
5.2.1 NTPとは 83
5.2.2 NTPの時刻同期の仕組み 84
5.2.3 RHEL6における時刻同期機能 88
5.2.4 システム時計の補正 88
5.3 NTP環境の設計・導入 91
5.3.1 NTP環境の設定手順 91
5.3.2 参照するNTPサーバの構成・参照設計 92
5.3.3 システム時計の時刻補正モード選定 93
5.3.4 RHELシステム起動時の初期参照NTPサーバの設計 94
5.3.5 ntpdの高度な動作設計 94
5.3.6 NTP環境の導入 95

5.4 NTP運用の設定....97
5.4.1 NTP設定ファイル....97
5.4.2 ntpdの起動オプション設定....103
5.4.3 ntpdateのオプション設定....104
5.4.4 時間補正関連ファイル（/etc/ntp/step-tickers）....105
5.4.5 補正値記録ファイル（/var/lib/ntp/drift）....105
5.5 NTP運用の確認点....105
5.5.1 NTP運用時の確認点....105
5.5.2 NTP運用の稼働状況の確認方法....106
5.6 うるう秒への対応....109
5.6.1 RHELにおけるシステム時計の調整方法....109

6 システム構築：RHELシステム環境構築....112
6.1 RHELシステムの環境構築概要....112
6.1.1 RHELシステムの環境構築手順....112
6.2 インストール前の準備....113
6.2.1 カスタマーポータルへのサブスクリプションの登録・更新....113
6.2.2 インストール媒体の作成....113
6.3 RHELのインストール....114
6.3.1 RHELのインストールにおける注意事項....114
6.3.2 RHELのインストール方法....115
6.4 RHELシステム環境設定....115
6.4.1 SAN環境におけるマルチパス構成....115
6.4.2 デバイス名の変更防止....116
6.4.3 時刻補正（NTP）....116
6.4.4 ファイルシステムオプション....116
6.4.5 OOM Killerの動作....116
6.5 レスキューモードの起動（トラブル発生時）....117
6.5.1 レスキューモード起動準備....118
6.5.2 レスキューモードの起動....118
6.5.3 レスキューモードの終了方法....119

7 運用・保守：バックアップ・リストア....120
7.1 バックアップ・リストアの概要....120
7.1.1 バックアップ・リストアの要件....120
7.1.2 バックアップの方法....120
7.1.3 バックアップデバイス....121
7.1.4 バックアップ前の確認事項....123
7.2 バックアップ前の準備....125
7.2.1 ランレベル1（シングルユーザーモード）への移行....125
7.2.2 ファイルシステムの整合性チェック....126
7.3 バックアップの実施....127
7.3.1 テープ装置を使用したローカルバックアップ....128
7.3.2 テープ装置を使用したリモートバックアップ....130
7.3.3 データカートリッジを使用したローカルバックアップ....133
7.3.4 データカートリッジを使用したリモートバックアップ....134
7.4 リストア前の準備....135
7.4.1 システムの起動....135

7.4.2 ディスク交換後の注意事項.....136
7.4.3 リストアするディスクパーティションのファイルシステム作成.....136
7.5 リストアの実施.....138
7.5.1 テープ装置を使用したローカルリストア.....138
7.5.2 テープ装置を使用したリモートリストア.....141
7.5.3 データカートリッジを使用したローカルリストア.....144
7.5.4 データカートリッジを使用したリモートリストア.....146
7.5.5 システムディスクパーティションリストア後の操作.....149

用語解説..... 151

本書をお読みになる前に

本書は、Red Hat Enterprise Linux 6をお使いになる方を対象に、設計／導入／運用・保守に関する情報や参考となる考え方を提供しています。導入から運用・保守をスムーズに実施するため、各段階において本書をご活用ください。

■ 対象機種

- FUJITSU Server PRIMEQUESTシリーズ

■ 本書の位置づけ

本書は、Red Hat 社から公開されているRed Hat Enterprise Linux 6向けマニュアルに対する補足情報を提供します。そのため、Red Hat Enterprise Linux 6の機能や使い方の詳細については、Red Hat社が公開するマニュアルをお読みください。

なお、仮想マシン機能（KVM）を利用される場合は、次のマニュアルをお読みください。

- 『Red Hat Enterprise Linux 6 Virtualization Administration Guide』
- 『Red Hat Enterprise Linux 6 Virtualization Getting Started Guide』
- 『Red Hat Enterprise Linux 6 Virtualization Host Configuration and Guest Installation Guide』

RHELゲストのインストールについては、「6 システム構築：RHELシステム環境構築」（→P.112）も併せてお読みください。

■ 高度な安全性が要求される用途への使用について

本製品は、一般事務用、パーソナル用、家庭用、通常の産業などの一般的用途を想定して開発・設計・製造されているものであり、原子力施設における核反応制御、航空機自動飛行制御、航空交通管制、大量輸送システムにおける運行制御、生命維持のための医療用機器、兵器システムにおけるミサイル発射制御など、極めて高度な安全性が要求され、仮に当該安全性が確保されない場合、直接生命・身体に対する重大な危険性を伴う用途（以下「ハイセイフティ用途」という）に使用されるよう開発・設計・製造されたものではありません。

お客様は本製品を必要な安全性を確保する措置を施すことなくハイセイフティ用途に使用しないでください。また、お客様がハイセイフティ用途に本製品を使用したことにより発生する、お客様または第三者からのいかなる請求または損害賠償に対しても富士通株式会社およびその関連会社は一切責任を負いかねます。

本書の構成

本書の構成と、各章の内容について説明します。

章	内容
1 システム設計：ハードディスク	この章では、ハードディスク関連の設計情報を説明します。
2 システム設計：ネットワーク	この章では、ネットワークの設計に関する設定方法や注意事項について説明します。

章	内容
3 システム設計：デバイス名ずれを考慮した設計	この章では、デバイス名ずれを考慮した対処方法について説明します。
4 システム設計：ログファイル運用	この章では、ログファイル運用について説明します。
5 システム設計：時刻補正の導入（NTP）	この章では、Linuxシステムにおける時刻の仕組みや時刻補正の必要性、時刻補正に重要な役割を果たすNTP（Network Time Protocol）サービスについて説明します。
6 システム構築：OS環境構築	この章では、OS環境構築の方法について説明します。
7 運用・保守：バックアップ／リストア	この章では、資産を確実に保護するために必要なバックアップ／リストアについて説明します。
用語解説	本文中で使用している専門用語について説明しています。

本書の表記

本書の中で記載している表記について説明します。

■ 本文中の記号

本文中では、次の記号を使用しています。

	RHELシステムの運用に影響するなど、重要な注意点について記述しています。
	理解を助けるための補足情報などを記述しています。
『　』	関連情報を記述したマニュアルを示しています。

■ 製品表記

本文中では、製品名称を次のように表記しています。

ソフトウェア製品名称	本書での表記	
Red Hat Enterprise Linux	RHEL	
Red Hat Enterprise Linux 6 (for Intel64)	RHEL6(Intel64)	RHEL6
Red Hat Enterprise Linux 6 (for x86)	RHEL6(x86)	
Red Hat Enterprise Linux 6.*n* (for Intel64)	RHEL6.*n*(Intel64)	RHEL6.*n* ［注1］
Red Hat Enterprise Linux 6.*n* (for x86)	RHEL6.*n*(x86)	

［注1］ ： "*n*"にはマイナーリリースを示す数字が入ります。

ハードウェア製品名称	本書での表記
FUJITSU Server PRIMEQUEST 1000シリーズ	PRIMEQUESTシリーズ

■ コマンド入力

本文中では、コマンド入力を次のように表記しています。

ユーザー可変（ユーザーの環境により異なる）の文字列

次のように、＜ ＞で括って表記します。

```
# e2label ＜デバイス名＞ ＜ラベル名＞
```

追加/変更対象文字列

次のように、太字で表記します。

```
NETWORKING=yes
HOSTNAME=＜ホスト名＞
・
・
VLAN=yes
```

説明を有する文字列

次のように、文字列の下に下線を引き、枠下に説明文を記載しています。

```
disk = ["phy:＜デバイス名＞,＜ラベル名＞,w"]
```

■ 容量の単位

データ容量を表す単位（KB,MB,GB,TB）として、本書では断りのない限り、1KB=1024byteで算出した値を表記しています。

■ キーの表記と操作方法

本文中のキーの表記は、キーボードに書かれているすべての文字を記述するのではなく、説明に必要な文字を次のように記述しています。

例：【Ctrl】キー、【Enter】キー、【→】キーなど

また、複数のキーを同時に押す場合には、次のように「＋」でつないで表記しています。

例：【Ctrl】+【F3】キー、【Shift】+【↑】キーなど

参照マニュアル

本文中で参照するマニュアルは、次のように『マニュアル名称』と表記しています。

■ PRIMEQUESTシリーズ

マニュアル名称	本書での表記
『PRIMEQUESTシリーズ 導入マニュアル』	『導入マニュアル』
『PRIMEQUESTシリーズ 運用管理マニュアル』	『運用管理マニュアル』

マニュアル名称	本書での表記
『PRIMEQUESTシリーズ システム構成図』 http://jp.fujitsu.com/platform/server/primequest/catalog/	『システム構成図』

■ Red Hat社から提供される公開マニュアル

Red Hat社のカスタマーポータル（https://access.redhat.com/home）において、RHEL6の製品マニュアル（英語版）が公開されています。RHEL6の利用方法に関するマニュアルは 次のとおりです。

マニュアル名称（本書での表記）	参照のタイミング	内容
『Red Hat Enterprise Linux 6 Release Note』（『Release Note』）	導入/構築	マイナーリリースに関する新機能および更新情報などが記載されています。メジャーバージョンの初版では、このバージョン特有の機能について記載されています。
『Red Hat Enterprise Linux 6 Deployment Guide』（『Deployment Guide』）	導入/構築	システムの設定およびカスタマイズ方法が記載されています。お読みになるには、システムの基礎知識が必要です。
『Red Hat Enterprise Linux 6 Installation Guide』（『Installation Guide』）	導入/構築	一般的なインストールについての関連情報が記載されています。 富士通製品の機種には、富士通が提供する独自のインストーラ（ServerView Installation Manager（以降、SVIMと表記します））が添付されています。SVIMを使用する場合は、『Installation Guide』に記載されているインストール方法は参照しません。
『Red Hat Enterprise Linux 6 Storage Administration Guide』（『Storage Administration Guide』）	導入/構築/運用・保守	ストレージ装置およびファイルシステムを、効果的に管理するための技術情報が記載されています。
『Red Hat Enterprise Linux 6 Logical Volume Manager Administration』（『Logical Volume Manager Administration』）	導入/構築/運用・保守	LVM（Logical Volume Manager）機能の使用方法について記載されています。
『Red Hat Enterprise Linux 6 DM Multipath』（『DM Multipath』）	導入/構築/運用・保守	DM-MP（Device-Mapper Multipath）機能を利用するための情報が記載されています。
『Red Hat Enterprise Linux 6 Security Guide』（『Security Guide』）	導入/構築/運用・保守	悪意ある活動からシステムを守るために必要なプロセスおよび習慣について、システム管理者および利用者向けの情報が記載されています。

マニュアル名称（本書での表記）	参照のタイミング	内容
『Red Hat Enterprise Linux 6 Virtualization Administration Guide』 『Red Hat Enterprise Linux 6 Virtualization Getting Started Guide』 『Red Hat Enterprise Linux 6 Virtualization Host Configuration and Guest Installation Guide』 （『Virtualization Guide』）	導入/構築/運用・保守	RHEL6が提供する仮想機能の設定、管理、およびトラブルシューティング情報が記載されています。
『Red Hat Network 証明書ベースのサブスクリプション管理』	導入/構築/運用・保守	Red Hatカスタマーポータルにおけるサブスクリプションとシステムの管理についての基本的な情報が記載されています。

本書は、最新の情報が反映されるマニュアルとして、英語圏向けのカスタマーポータルで公開されるマニュアルを参照対象とします。なお、日本語対応マニュアルは、日本語向けのカスタマーポータルで確認してください。

- RHEL6の製品マニュアル（英語版）を参照するためには、カスタマーポータルの言語の選択で「English」を選択し、英語圏向けのカスタマーポータルに表示を切り替えます。英語圏向け以外のカスタマーポータルでは、RHEL6の製品マニュアルが配置されていない場合があります。

■ Red Hat社のナレッジベース

Red Hat社のカスタマーポータル（https://access.redhat.com/home）のナレッジベースでは、利用者からよくある質問と回答を公開しています。トラブルが発生した場合や、困ったときなど、参考情報があるかどうか確認してください。

- ナレッジベースを利用の際は、事前にRed Hat社製品のサブスクリプション登録が必要です。サブスクリプション登録については、「6.2.1 カスタマーポータルへのサブスクリプションの登録・更新」（→P.113）をお読みください。

■ オンラインマニュアル（manコマンド）

manコマンドは、オンラインマニュアルを参照するコマンドです。manコマンドを使用して、コマンド、関数、設定ファイルなどの使用方法を確認できます。

manコマンドの詳細説明については、manコマンドのオンラインマニュアルを参照してください。

書式

```
man [オプション] [セクション] キーワード
```

キーワードには、コマンド名、関数名、設定ファイル名などを指定します。

セクション

オンラインマニュアルは、次のセクション（章）に分かれています。

章番号	内容
1	実行プログラムまたはシェルのコマンド
2	システムコール（カーネルが提供する関数）
3	ライブラリコール（システムライブラリに含まれる関数）
4	スペシャルファイル（通常/devに置かれている）
5	ファイルのフォーマットとその規約
6	ゲーム
7	マクロのパッケージとその規約
8	システム管理用のコマンド
9	カーネルルーチン

保守サービスについて

富士通は、有償サポートサービス（SupportDesk Standard）を提供しています。
有償サポートサービスでは、RHELに関する質問、インストール、運用などで発生する疑問やトラブルなどについて、富士通サポートセンター（OSC：One-stop Solution Center）が一括対応いたします。

また、この有償サポートサービスには、RHELを利用する際に必要なサブスクリプション（利用権）が含まれています。サブスクリプションは、RHELを利用する権利であり、Red Hat社からインストールイメージ（ISOファイル形式）、セキュリティアップデート、およびその他アップデートを入手するための権利を含みます。

RHELの使用に際しては、有償サポートサービスの契約を推奨します。

有償サポート・サービス（SupportDesk Standard）については、次のサイトを参照してください。

http://jp.fujitsu.com/solutions/support/sdk/sd-standard/linux/

商標および著作権について

Red Hat、Shadowman logo、JBossは米国およびそのほかの国において登録された Red Hat, Inc. の商標です。

Linuxは、Linus Torvalds氏の登録商標です。

Intel、インテル、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Atom、Intel Atom Inside、Intel Core、Core Inside、Intel vPro、vPro Inside、Celeron、Celeron Inside、Itanium、Itanium Inside、Pentium、Pentium Inside、Xeon、Xeon Phi、Xeon Inside、Ultrabookは、アメリカ合衆国および/またはその他の国におけるIntel Corporationの商標です。

その他の各製品名は、各社の商標、または登録商標です。 その他の各製品は、各社の著作物です。

Copyright FUJITSU LIMITED 2010-2014

改版履歴

版数	変更日	変更箇所	修正概要
初版	2010年11月17日	全章	初版作成
2版	2011年6月1日	2章、4～8章	RHEL6.1対応
3版	2011年12月20日	2～8章	RHEL6.2対応
4版	2012年12月20日	1 章、3 章、5 ～ 8 章	RHEL6.3対応
5版	2013年3月7日	全章	構成変更、RHEL6.4対応
6版	2013年12月19日	全章	RHEL6.5対応

1 システム設計：ハードディスク

ここでは、ハードディスク関連の設計情報を説明します。

1.1 ディスクパーティション

ここでは、ディスクパーティションの作成について説明します。

- ディスクパーティションについては、Red Hat社のマニュアル『Installation Guide』を参照してください。

1.1.1 パーティションテーブル

パーティションテーブルは、ディスクパーティションの位置や大きさを管理するために、ハードディスクの先頭領域に存在する管理テーブルです。パーティションテーブルは、partedコマンドなどを使用して作成します。
RHELがサポートするパーティションテーブルの種類および特徴は、次のとおりです。

表 1 : パーティションテーブルの種類と特徴

種類	特徴
MS-DOS形式（またはMBR（Master Boot Record）形式）	• パソコンで採用されている形式です。 • ディスクパーティションは最大4個まで、次の構成が可能です。 - 基本パーティション×4 または - 基本パーティション×3、拡張パーティション×1 • 拡張パーティション内は、さらに論理パーティションで分割できます。 • SCSIディスクの場合、ディスクパーティションは最大15個までの構成が可能です（基本パーティション3個＋拡張パーティション1個とした場合、論理パーティションの数は15－4＝11個）。 • 2TBを超える領域にディスクパーティションを作成できません。
GPT（GUID Partition Table）形式	• EFI標準規格の一部として提唱された形式です。 • ディスクパーティションは最大15個（RHELがサポートする最大数）の構成が可能です。 • 2TBを超える領域にディスクパーティションを作成できます。

パーティションテーブルの形式は、RHELを起動するファームウェアと次のように組み合わせます。

- パーティションテーブルのディスクラベルは、パーティションテーブルがMS-DOS形式ならばmsdos、GPT形式ならばgptを設定します。

表 2 : ファームウェアとパーティションテーブルの組合せ

ファームウェアからのOS起動方法	対応アーキテクチャ	使用できるパーティションテーブルの種類	
		システムディスク	データディスク
UEFIのBIOS互換モード	x86 Intel64	MS-DOS形式 GPT形式［注1］	MS-DOS形式 GPT形式
UEFIモード	Intel64	GPT形式	

［注1］：RHELインストーラは、パーティションテーブルが存在しない場合、MS-DOS形式でパーティションテーブルを作成します。そのため、RHELインストール時、あらかじめGPTパーティションテーブルが作成されていなければなりません。

ファームウェアの選択指針については、「6.3.1 RHELのインストールにおける注意事項」（→P.114）をお読みください。

1.1.2 ディスクパーティションの見積り方法

ここでは、OSの動作に必要なディスクパーティションの容量見積りについて説明します。

- /（root）パーティション
- /bootパーティション
- /boot/efiパーティション
- swapパーティション

- /boot/efiパーティションは、ファームウェアがUEFIモードの場合に必要となります。
- ミドルウェアのインストール、メモリダンプ機能の設定などを行う場合、それぞれの領域要件を考慮して、ディスクパーティションの構成および容量を決定してください。

■ ディスクパーティションの容量見積り

ディスクパーティションの容量を見積る場合、ディスクパーティションの用途および容量の目安を考慮します。

- ファイルシステムの管理領域として、ディスクパーティション領域の数%が使用されますので、容量は余裕をもって見積ります。
- ディスクパーティションの容量は、1KB=1024byteで計算します。

  例：73GBの容量を確保する場合

  次の領域が必要となります。

  ```
  73×1024×1024×1024＝78,383,153,152byte
  ```

  ディスク容量は1KB=1000byteで表記されます。このため、ディスク操作コマンドが扱う「73GBのディスク」の実際の容量は73,000,000,000byteであり、必要な容量を満たしません。この例のように、73GBの容量を確保する場合、少なくても78.4GB以上の容量を持つディスクを用意する必要があります。

表 **3 :** ディスクパーティションの見積り（**1KB=1024byte**）

ディスクパーティション	用途	推奨容量	推奨ファイルシステム
/	rootディレクトリ	20GB以上	ext3
/boot	ブートストラップのプロセス中に使用されるファイルと共に、OSのカーネルが格納される領域	256MB	ext3
/boot/efi	ファームウェアがUEFIモードの場合、ブートストラップのプロセスで使用されるファイルが格納される領域	256MB	vfat
swap	現在動作中のプロセスを取り扱うだけの十分なシステムメモリがないとき、仮想記憶域として使用される領域	システムメモリの容量に応じて、推奨容量が異なります。	swap領域は独自の形式のため、ファイルシステムは不要です。

- ファイルシステムの詳細については、「1.2 ファイルシステム」（→P.18）をお読みください。
- RHELインストーラは、ディスクパーティションの容量が大きい順序でディスクパーティション番号を割り振ります。そのため、RHELインストール時にディスクパーティションを作成する場合、ディスクパーティション番号の割振りは、ディスクパーティションを作成した順序と一致しないことがあります。

■ /（root）パーティション容量見積りの注意事項

20GB以上の容量を確保します。ただし、次の2つの要因により、ディスク容量が増加することを考慮して決定します。/usrパーティションおよび/optパーティションを設定している場合も同様です。

- ソフトウェアの追加または修正適用
  次の作業により、ファイルシステムの使用量が増加する可能性があります。
  - ISVのソフトウェア、オプションソフトウェアの追加
  - OSのソフトウェアグループの変更
  - errataの適用など
- ログファイルの保存
  /varディレクトリには、メールボックスやプリンタスプールのほかに、RHELやミドルウェアのログファイルが保存されます。ログファイルは、保存期間が長くなることで、予想もしない大きさとなることがあります。そのため、ログファイルを退避する運用または十分な容量を確保してください。

■ swapパーティション容量見積りの注意事項

swapパーティションの容量は、システムにかかるメモリ負荷に応じて設定します。システム導入時にメモリ負荷を見積ることが難しい場合は、次表を参考にして、サーバのRAM（物理メモリ）容量に応じた大きさに設定してください。

表 **4 : swap**パーティション容量（**1KB=1024byte**）

RAM容量（**GB**）	最低限必要な**swap**領域の容量（**GB**）	
	hibernationを使用しない場合	**hibernation**を使用する場合［注1］
M ≦ 2	Mの2倍	Mの3倍
2 < M ≦ 8	Mと同量	Mの2倍
8 < M ≦ 64	Mの0.5倍	Mの1.5倍
64 < M	4	4

M：RAM（物理メモリ）容量（GB）

［注1］：RHEL6.3以降で使用できます。

- swapパーティションの設計に関する詳細は、Red Hat社のマニュアル『Installation Guide』を参照してください。

1.1.3 ディスクパーティションを作成するコマンド

ディスクパーティションを作成するには、主にpartedコマンドまたはfdiskコマンドが使用されます。
それぞれのコマンドの特徴は、次のとおりです。なお、ディスクパーティションの作成は、GPT形式およびMS-DOS形式に対応したpartedコマンドの使用を推奨します。

表 **5 :** コマンドの特徴

コマンド名	特徴
parted	MS-DOS形式／GPT形式のパーティションテーブルを操作可能です。 GPT形式では、2TBを超える領域にディスクパーティションを作成できます。
fdisk	MS-DOS形式のパーティションテーブルを操作可能です。

partedコマンドの使い方は、オンラインマニュアルまたはRed Hat社のマニュアル『Storage Administration Guide』を参照してください。

fdiskコマンドの使い方は、オンラインマニュアルを参照してください。

1.1.4 partedコマンド使用時の注意事項

partedコマンド使用時の注意事項を説明します。

■ サブコマンドの使用制限

partedコマンドは、mkfsサブコマンド、mkpartfsサブコマンド、cpサブコマンド、resizeサブコマンドを提供します。しかし、これらのサブコマンドを使用した場合、正しい動作をしない場合があります。そのため、サブコマンドは使用せずに、代替手段で対応してください。

表 **6 :** サブコマンドの機能と代替手段

サブコマンド	機能	代替手段
mkfs	ディスクパーティション上にファイルシステムを作成します。	mke2fsコマンドによりファイルシステムを作成します。
mkpartfs	ディスクパーティションとファイルシステムを作成します。	mkpartサブコマンドを使用してディスクパーティションを作成したあと、mke2fsコマンドによりファイルシステムを作成します。
cp	コピー元のディスクパーティション内のファイルシステムを、コピー先のディスクパーティションにコピーします。	mkpartサブコマンドを使用してディスクパーティションを作成し、mke2fsコマンドによりファイルシステムを作成したあと、ファイルやディレクトリをコピーします。
resize	既存のディスクパーティションの領域を拡張します。	該当ディスクパーティションをバックアップし、mkpartサブコマンドを使用してディスクパーティションを作成し直します。その後、mke2fsコマンドによりファイルシステムを作成し、バックアップをリストアします。

■ ディスクパーティション作成時の容量単位

ディスクパーティションの操作時の容量は、単位「MiB」（1MiB＝1×1024×1024［byte］）の使用を推奨します。

- unitサブコマンドで容量単位をMiBに変更しない場合、partedコマンドは次の単位をパーティションサイズの指定および表示に使用します。
  - 1KB＝1×1000［byte］
  - 1MB＝1×1000×1000［byte］
  - 1GB＝1×1000×1000×1000［byte］

■ ディスクパーティションの作成例

ディスクパーティション5まで作成されている/dev/sdaの空き領域に、100MiB（100MiB＝100×1024×1024［byte］）のディスクパーティション6を追加する場合の例です。

1. partedコマンドを実行します。

```
# parted /dev/sda
```

2. unit MiBサブコマンドを使用して、容量単位を切り替えます。

```
(parted) unit MiB
(parted) print
モデル: SEAGATE ST373307LC (scsi)
ディスク /dev/sda: 70008MiB
セクタサイズ (論理/物理): 512B/512B
パーティションテーブル: gpt

 番号 開始 終了 サイズ ファイルシステム 名前 フラグ
（略）
 5 763MiB 20295MiB 19532MiB
```

3. ディスクパーティション容量に合わせた開始位置および終了位置を算出します。

容量＝ 100［MiB］

開始位置＝（番号 5 の終了）＝ 20295［MiB］

終了位置＝（開始）+（サイズ）＝ 20295 ＋ 100 ＝ 20395［MiB］

4. ディスクパーティションを作成します。

mkpartサブコマンドを使用して、空き領域にディスクパーティションを作成します。

```
(parted) mkpart
パーティションの種類? logical/論理?
ファイルシステムの種類? [ext2]?
開始? 20295 ←手順3で計算した開始位置を指定
終了? 20395 ←手順3で計算した終了位置を指定
```

5. 作成したディスクパーティションを確認します。

printサブコマンドを使用して、意図したディスクパーティションが作成されたことを確認します。

```
(parted) print
（略）
番号 開始 終了 サイズ ファイルシステム 名前 フラグ
（略）
5 763MiB 20295MiB 19532MiB
6 20295MiB 20395MiB 100MiB
```

- fdiskコマンドを使用する場合、ディスクパーティション終了位置指定の「K」、「M」、および「G」の単位は、1000byte単位です。

1.2 ファイルシステム

ファイルシステムに関する補足情報について説明します。

ファイルシステムには複数の種類があり、ディスクパーティションごとに作成します。

ファイルシステムをRHELシステムに組み込んで利用可能にすることを「ファイルシステムをマウントする」といいます。ファイルシステムをマウントするためには、/etc/fstabファイルに記述するか、またはmountコマンドを実行します。

- ファイルシステムの詳細およびファイルシステム種別（ext3、ext4など）ごとの特徴については、Red Hat社のマニュアル『Storage Administration Guide』をお読みください。

1.2.1 ファイルシステムの最大値

サポートするファイルシステムの最大値について説明します。

表 **7 :** ファイルシステムの最大値

ファイルシステム種別	ファイルシステムの最大値	ファイルサイズの最大値
ext2	8TB	2TB
ext3	16TB［注1］	2TB
ext4	16TB	16TB［注2］

［注1］ : ext3ファイルシステムのファイルシステムの最大値は、RHEL6.0までは8TB、RHEL6.1以降は16TBです。

［注2］ : ext4ファイルシステムの最大ファイルサイズは、アーキテクチャがIntel64の場合です。アーキテクチャがx86の場合、ext4ファイルシステムの最大ファイルサイズは8TBです。

- ext2ファイルシステムおよびext3ファイルシステムの場合、論理的な最大ファイルサイズは2TBですが、実際に作成可能なファイルサイズは次表のとおりです。

  表 **8 :** 実際に作成可能なファイルサイズ

カーネルバージョン	最大ファイルサイズ
RHEL6以降	2,194,719,883,264（0x1feff7fc000）byte
RHEL5以前（参考）	2,196,875,759,616（0x1ff7fffd000）byte

1.2.2 ファイルシステム使用上の注意事項

ファイルシステムを使用する場合の注意事項を説明します。

■ 推奨するファイルシステム

富士通は、RHEL6からext4ファイルシステムを新たにサポート対象としましたが、これまでの実績を踏まえ、品質面・安定性を重視する場合は、ファイルシステム（特にシステムボリューム）に、ext3ファイルシステムの使用を推奨します。

■ ext4ファイルシステムを使用する場合の注意点

ext4ファイルシステム固有の問題が発生した場合、ファイルシステムをext3ファイルシステムに変更することで回避することができます。ext3ファイルシステムに変更するためには、ext4ファイルシステムをext3ファイルシステムの上限値の範囲内で使用する必要があります。

ext4ファイルシステムをext3ファイルシステムに変更する手順は次のとおりです。

1. dumpコマンドなどを使用して、ext4ファイルシステムのバックアップを採取します。
2. ディスクパーティションをext3ファイルシステムで再構成します。
3. restoreコマンドなどを使用して、ext3ファイルシステムにバックアップをリストアします。

ext3ファイルシステムをext4ファイルシステムに変更する場合は、上記手順と同様に操作してください。

■ データ書込み同期の考慮

ファイルシステムは、データへのアクセス速度向上のために、ページキャッシュと呼ばれるメモリ上のキャッシュ機構を使用します。通常、アプリケーションからのデータ書込みは、このページキャッシュへの書込み終了時点で完了します（その後、非同期にページキャッシュからディスクなどの媒体への書込み動作が行われます）。

このため、ハードウェアの故障や停電などにより、予期しないシステム停止が発生すると、データの不整合が発生する場合があります。ディスクなどの媒体への書込み動作を必要とするデータの場合、fsyncシステムコールなどを使用して、ページキャッシュとディスクなどの媒体上のデータとの同期をとるようアプリケーションを設計してください。

- closeシステムコールを使用しても、ページキャッシュとディスクなどの媒体上のデータとの同期処理は行われません。そのため、アプリケーションによる積極的なデータの同期処理が重要です。

1.2.3 ファイルシステムのマウントオプション

ファイルシステムをマウントする際に使用するオプションにより、ファイルシステムの動作を変更できます。オプションは、ファイルシステム種別に共通するオプションと、ファイルシステム種別固有のオプションがあります。

ここでは、ext3ファイルシステムおよびext4ファイルシステム固有のオプションのうち、設定を推奨するオプションと注意が必要なオプションについて説明します。

- ファイルシステムのオプションについては、mountコマンドのオンラインマニュアルを参照してください。

■ [data]オプションの運用

[data]オプションは、ファイルシステムのジャーナル機能を制御します。デフォルトはorderedモード（data=ordered指定）で動作します。特別な理由がない限り、デフォルト設定での運用を推奨します。なお、journalモード（data=journal指定）では、openシステムコールのO_DIRECTフラグをサポートしていません。journalモードを設定しているファイルシステムで、O_DIRECTフラグを指定したプログラムを実行すると、openシステムコールが不正な引数としてエラーとなります。O_DIRECTフラグを指定したプログラムの実行が必要な場合は、ファイルシステムのモードをjournalモード以外に設定してください。

■ [data_err]オプションの運用

[data_err]オプションは、ファイルシステムがorderedモードで使用されている場合に機能します。[data_err]オプションは、次の設定から選択します。

- data_err=ignore
  データ書出し時にジャーナル処理でエラーが発生した場合、エラーメッセージを出力するだけで処理を継続します。
- data_err=abort
  データ書出し時にジャーナル処理でエラーが発生した場合、I/Oエラーを通知します。

デフォルトの動作はdata_err=ignoreですが、I/Oエラーを早期に検出するために、data_err=abortの設定を推奨します。

1.3 ディスクの冗長化

LVM、MDまたはDM-MPを利用したディスク管理をすることで、フレキシブルなストレージ管理が可能となり、かつデータの可用性と信頼性が向上します。

1.3.1 LVM

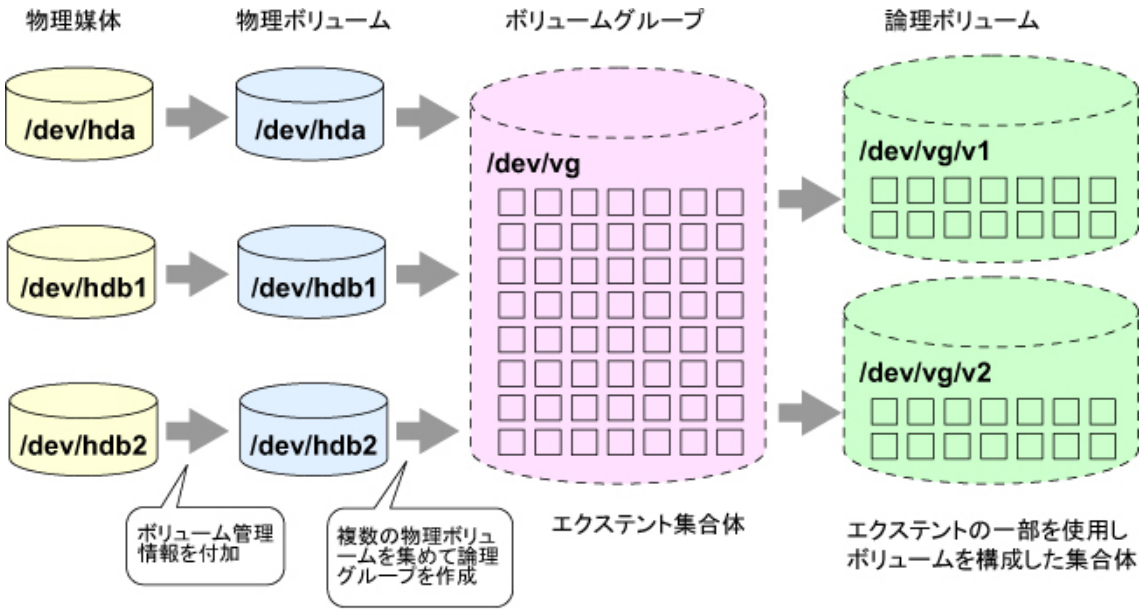
LVMは、ディスクの容量などの物理的制約を受けることのない、論理的なディスクを扱うための機能です。

- LVMの詳細については、Red Hat社のマニュアル『Logical Volume Manager Administration』をお読みください。

LVMの特徴は次のとおりです。

- 複数のドライブから成る物理的な記憶装置をまとめ、論理的な記憶装置（論理ボリューム）を作成できます。
- 論理ボリュームは必要に応じて記憶装置を増設または減設できます。
- 論理ボリューム上に作成したディスクパーティションは拡張または縮小できます。
- 論理ボリューム上に作成したディスクパーティションのスナップショットを作成できます。

図 **1 : LVM**のイメージ



■ LVM構成要素

LVMの構成要素の名称と役割を説明します。

表 **9 : LVM**構成要素

構成要素	略称	説明
物理媒体	なし	ディスクまたはディスクパーティション（例：/dev/sda、/dev/sda1）。 ディスクパーティションをLVMの物理媒体として使用する場合には、ディスクパーティションタイプを「Linux LVM (8e)」とします。
物理ボリューム	PV	ボリューム管理情報を付加した物理媒体。 物理ボリュームを作成することで、初めてLVMで使用可能な領域（エクステント）を確保できます。
物理エクステント	PE	物理ボリュームを一定のサイズで区切った領域。
論理エクステント	LE	物理エクステントと全く同じもの。 物理ボリューム上のエクステントを物理エクステント、論理ボリューム上のエクステントを論理エクステントと呼びます。
ボリュームグループ	VG	複数の物理ボリュームを集めて作成する論理グループ。 この中からディスクパーティションとして使用する論理ボリュームを作成することができます。エクステントのサイズは、ボリュームグループ作成時に指定します。

構成要素	略称	説明
論理ボリューム	LV	ボリュームグループからエクステントを集めて作成する論理的なパーティション。 通常のファイルシステムと同様にフォーマットおよびマウントして使用します。

■ LVMボリューム状態

表 10 : LVMボリューム状態

ボリューム状態	説明
アクティブ	ボリューム情報がカーネル空間に存在し、ストレージとして利用可能な状態です。この状態では、論理ボリュームの構成変更はできません。
インアクティブ	ボリューム情報がユーザーランド（ユーザー空間）に存在し、論理的なボリュームの新規作成や修正作業が可能な状態です。LVM関連コマンドは一部を除き、アクティブ／インアクティブの状態を自動で変更します。

1.3.2 MD

MDは、ドライバ層でRAIDを実現するソフトウェアRAID機能です。MDを使用することで、複数のディスクを組み合わせて、ディスク故障時にデータの損失防止（ディスク冗長化）やディスクアクセスの処理速度の向上を実現します。

- MDの詳細については、Red Hat社のマニュアル『Storage Administration Guide』およびmd／mdadmコマンドのオンラインマニュアルを参照してください。

MDは、次のRAIDレベルをサポートします。

表 11 : MDでサポートするRAIDレベル

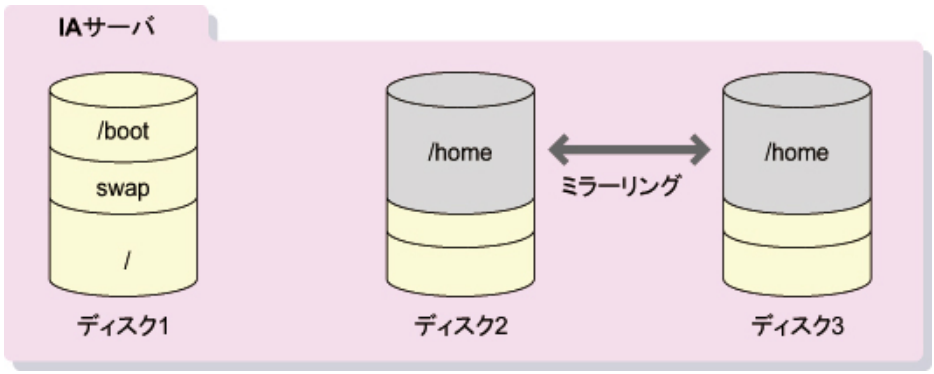
RAIDレベル	機能
linear	複数のディスクを使用して、1つの大容量の仮想ドライブを構成します。
RAID0	ストライピングとも呼ばれ、複数のディスクにデータを分割して書き込みます。I/O性能は向上しますが、冗長性がありません。
RAID1	ミラーリングとも呼ばれ、複数のディスクに同じデータを書き込みます。I/O性能は向上しませんが、冗長性があります。
RAID1+0	RAID0とRAID1を組み合わせた形態です。I/O性能が向上し、冗長性もあります。

- RAIDレベルの詳細については、次のサイトを参照してください。
  http://storage-system.fujitsu.com/jp/term/raid/

MDは、データボリューム向けのRAID構成をサポートします。データボリューム向けのRAID構成は、mdadmコマンドで作成します。

/homeパーティションをRAID1で構成し、同じサイズのディスクパーティションをディスク2、ディスク3に配置した例を次図に記します。

図 2：MDの構成



1.3.3 DM-MP

DM-MPは、マルチパス接続されたブロックデバイスを制御する機能です。DM-MPを使用したマルチパス機能では、冗長性を持たせたディスクアレイ装置へのアクセスパスを統合して、アクセスパス障害の発生に備えたシステム運用の継続性を実現します。

参考

- DM-MPの詳細については、Red Hat社のマニュアル『DM Multipath』をお読みください。

注意

- 「ETERNUSマルチパスドライバ」など、DM-MPと同等の機能を提供する製品をお使いの場合は、DM-MP機能は併用しないでください。

SAN環境においてDM-MPを利用することで、ディスクデバイスへのアクセスパスを簡単にマルチパスで構成できます。DM-MPは、次図のどちらの構成もサポートしています。経路となるアクセスパスは複数本で構成しておく必要があります。

図 3：ファイバーチャネルスイッチを経由するSAN構成

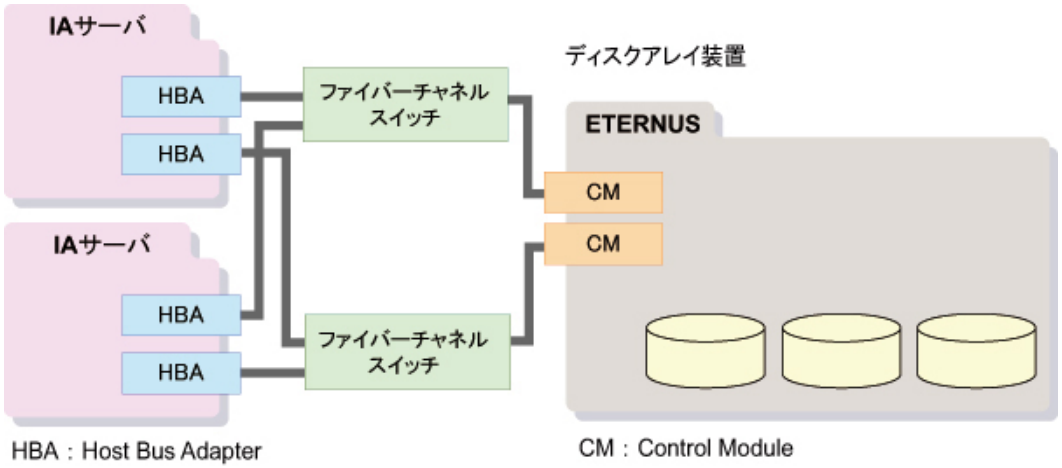
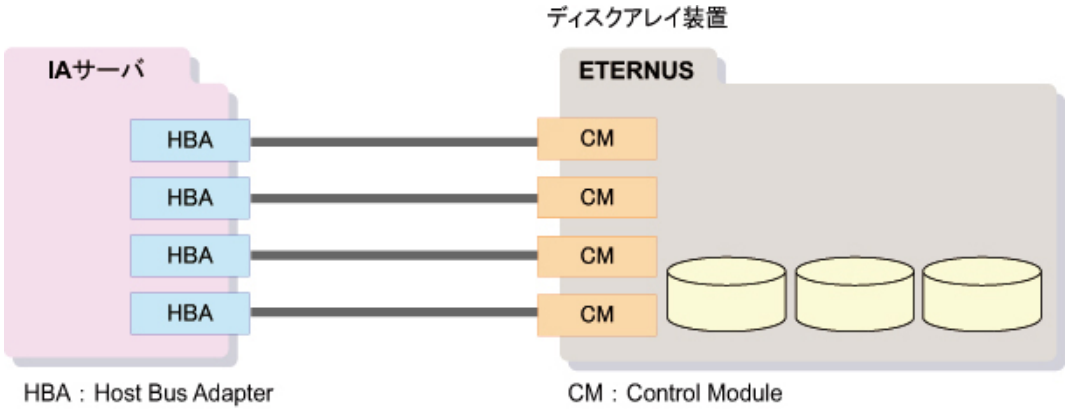


図 **4 :** ファイバーチャネルケーブルをサーバに直結する構成



■ マルチパス構成のデバイスファイル

マルチパス構成にしたディスクデバイスは、次のデバイスファイル名でアクセスできます。

表 **12 :** マルチパス構成のデバイスファイル

種別	デバイスファイル名	例：**alias**に**mpath0**を指定した場合
ボリューム	/dev/mapper/＜NNNNN＞［注1］	/dev/mapper/mpath0
パーティション	/dev/mapper/＜NNNNN＞pM［注1］［注2］	/dev/mapper/mpath0p1

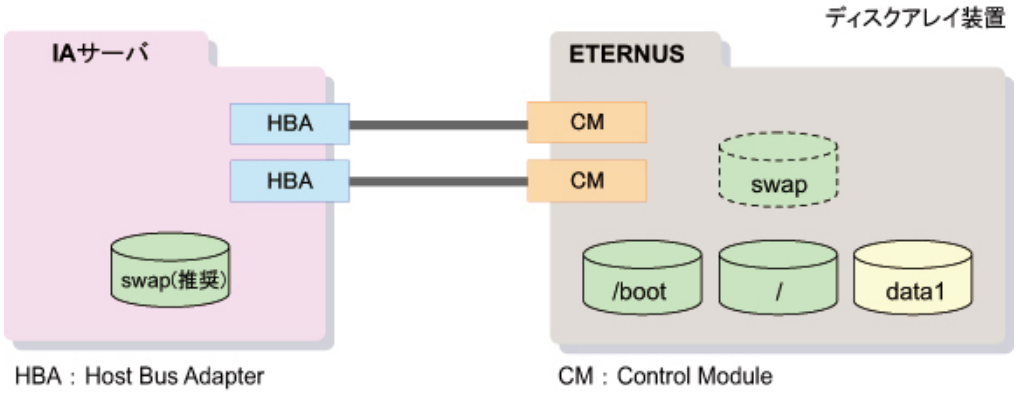
［注1］：NNNNNはボリュームを表す名前（alias名）を示します。

［注2］：pM（文字列'p' + 数字）はパーティション番号を示します。数字は自動的に割り振られます。

■ SAN Boot（システムボリューム）のマルチパス構成

システムボリュームをディスクアレイ装置上に配置するSAN Bootの構成をマルチパスで構成できます。

図 **5 : SAN Boot**（システムボリューム）のマルチパス構成



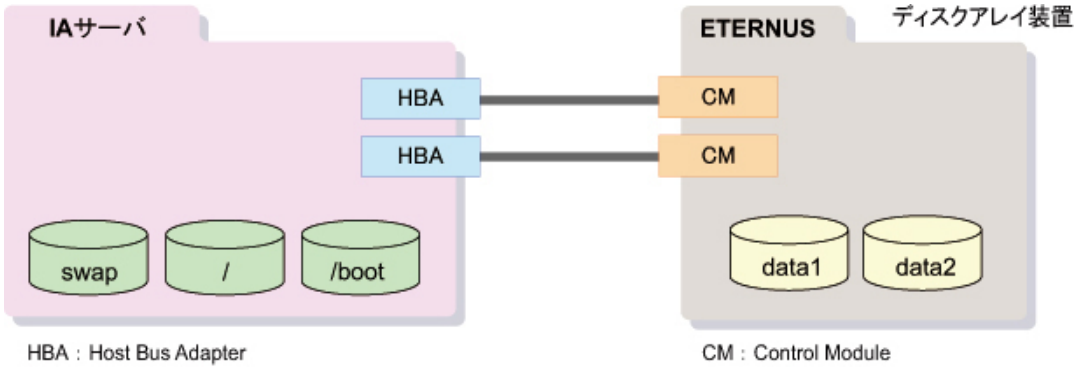
注 意

- swapパーティションは頻繁にアクセスするため、内蔵ディスクに配置することを推奨します。
- /bootファイルシステムおよび/（ルート）ファイルシステムを含むシステムボリュームは、ディスクアレイ装置に配置してください。

■ データボリュームのマルチパス構成

ディスクアレイ装置上にデータボリュームを配置して、マルチパスで構成することができます。

図 6 : データボリュームのマルチパス構成



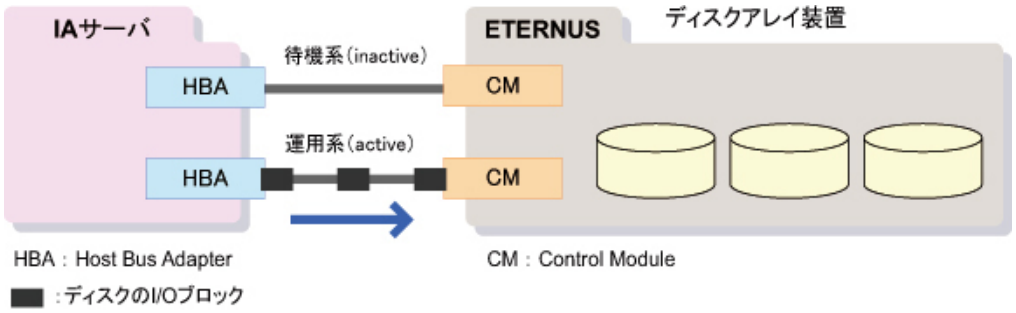
■ 運用待機方式（failover）

アクセスパスの1つを運用系、残りを待機系として運用する方式です。

次図は、一方を運用系（active）、残りのアクセスパスを待機系（inactive）とした場合のディスクI/Oの流れです。

運用系パスで故障が発生すると、待機系パスの1つが自動的に運用系（active）へ切り替わります。

図 7 : 運用待機方式（**failover**）



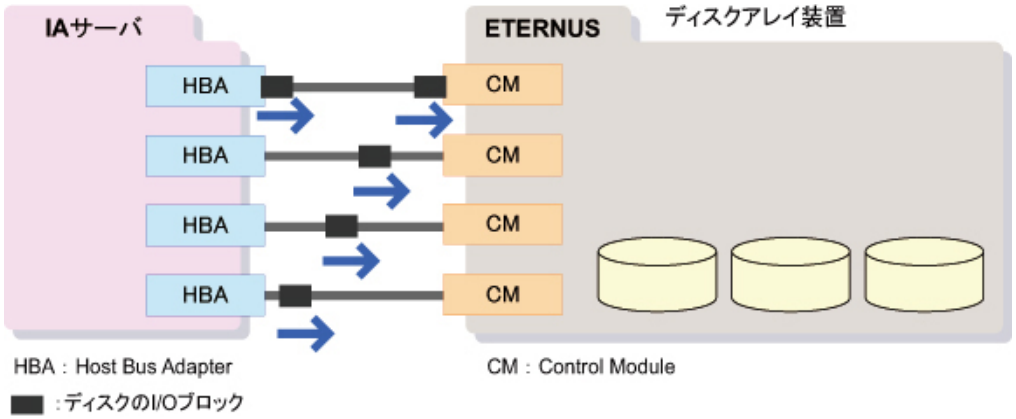
■ 負荷分散方式（multibus）

すべてのアクセスパスを使用して、ディスクI/Oのブロックをラウンドロビンで負荷分散することで、アクセス性能を向上させる方式です。

次図は、4つのパスすべてを使用して負荷分散した場合のディスクI/Oの流れ（ラウンドロビン）です。

アクセスパスで故障が発生すると、残りのアクセスパスを使用して運用を継続します。

図 8 : 負荷分散方式（multibus）



1.4 ディスク利用時の考慮点

ディスク利用における考慮点について説明します。

1.4.1 450台を超えるディスク

RHELシステムに450台を超えるディスクを接続する場合、udev処理のタイムアウト時間を長くする必要があります。

■ 詳細情報

RHELシステムに450台を超えるディスクを接続する場合、システム起動時に次のメッセージが表示されることがあります。

```
Starting udev: Wait timeout. Will continue in the background.[FAILED]
```

このメッセージは、udev機能がディスクを認識してデバイスファイルを作成しようとしたが、設定されたタイムアウト時間内にすべてのディスクを認識できなかったことを示します。

udev機能は、システムが認識したデバイスに対して、/devディレクトリ配下に対応するデバイスファイルを動的に作成する機能です。タイムアウト時間は、デフォルトで180秒となっているため、450台を超えるディスクを接続する場合、デバイスファイルの作成処理が180秒を超えることがあります。

■ 対処方法

RHELシステムに450台を超えるディスクを接続する場合、次の手順でudev処理のタイムアウト時間を変更します。

1. udev処理のタイムアウト時間を見積ります。
   RHELシステムに接続されているディスクの台数を確認し、次の式からudev処理のタイムアウト時間を見積ります（小数点以下は切上げ）。

   ```
   タイムアウト時間＝0.4×ディスク数
   ```

2. udev処理のタイムアウト時間を設定します。

/boot/grub/grub.conf（ファームウェアがUEFIモードの場合は/boot/efi/EFI/redhat/grub.conf）を編集し、「kernel」行に次のカーネルブートオプションを追加します。

```
udevtimeout = N
```

Nは手順1で求めた値（整数）です。

例：RHELシステムに1024台のハードディスクが接続されている場合

```
title Red Hat Enterprise Linux Server (2.6.32-71.el6)
      root (hd0,0)
      kernel /vmlinuz-2.6.32-71.el6 ro root=LABEL=/ rhgb quiet
udevtimeout=410
initrd /initramfs-2.6.32-71.el6.img
```

1.4.2 ディスクパーティション作成時のデバイスファイル

ディスクパーティション作成直後のデバイスファイルについて説明します。

■ 詳細情報

udev機能は、RHELシステム起動時に認識したデバイスに対するデバイスファイルだけを作成します。RHELシステム起動後にディスクパーティションを作成した場合、対応するデバイスファイルは、ディスクパーティション作成処理実行時にudev機能によって動的に作成されます。

udevによるデバイスファイル作成処理は、ディスクパーティション作成処理とは非同期に行われます。そのため、ディスクパーティション作成処理が完了しても、デバイスファイルが存在しない状態となる場合があります。

■ 対処方法

新たに作成したディスクパーティションにアクセスする場合は、当該ディスクパーティションに対応するデバイスファイルの作成が完了するまで待ちます（参考値：約1秒）。具体的な待ち時間は環境によって異なりますので、lsコマンドを実行することでデバイスファイルの作成完了を確認します。

例：デバイスファイルの作成が完了していない場合

```
# ls /dev/sda10
ls: cannot access /dev/sda10: No such file or directory
```

例：デバイスファイルの作成が完了している場合

```
# ls /dev/sda10
/dev/sda10
```

1.4.3 ディスクの活性増設とUSBドライブ

ディスクを活性増設する場合、デバイス名ずれ対策のため、USBフロッピーディスクドライブを取り外してから作業を開始します。

■ 詳細情報

USBフロッピーディスクドライブ（USB-FDD）を接続した状態でディスクを活性増設すると、増設したディスクのデバイス名は、USBフロッピーディスクドライブよりも後になります。そのため、USBフロッ

ピーディスクドライブを取り外した状態でRHELシステムを再起動すると、増設したディスクのデバイス名がずれます。

デバイス名の発生過程は、次のとおりです。

1. ディスクの活性増設前
   sda：HDD
   sdb：USB-FDD
2. ディスクの活性増設後
   sda：HDD
   sdb：USB-FDD
   sdc：増設したHDD ←増設したディスクのデバイス名はUSB-FDDより後
3. USBフロッピーディスクドライブの取外し後
   sda：HDD
   sdc：増設したHDD
4. 再起動
   sda：HDD
   sdb：増設したHDD ←増設ディスクのデバイス名がずれる

■ 対処方法

- RHELシステム起動時は、常にUSBフロッピーディスクドライブを取り外しておいてください。
- USBフロッピーディスクドライブは、使用後は速やかに取り外してください（接続したままで作業を継続しないでください）。umountコマンドにて、フロッピーディスクをアンマウントしてから、USBフロッピーディスクドライブを取り外してください。

- デバイス名ずれの詳細については、「3 システム設計：デバイス名ずれを考慮した設計」（→P.48）をお読みください。

1.4.4 rawデバイス

rawデバイスは、廃止される予定（時期は未確定）です。そのため、rawデバイスは使用しないことを推奨します。

■ 詳細情報

RHELシステムは、キャッシュメモリを介さずに直接ディスクデバイスにアクセスする手法としてrawデバイス機能を提供しています。しかし、本機能はアプリケーションの互換性維持のために提供された機能であり、RHELでは非推奨とされています。

■ 対処方法

rawデバイスの利用に対する代替手段として、openシステムコールのO_DIRECTフラグを用いたディスクデバイスファイルのオープンを推奨します。

アプリケーションプログラムが、ディスクデバイスに対してO_DIRECTフラグを立ててopenシステムコールを発行することにより、ディスクデバイスへのアクセスがキャッシュメモリを介さずに行われるようになります。

2 システム設計：ネットワーク

ここでは、ネットワークに関する設定方法と注意事項について説明します。

2.1 LANの構成

サーバが接続するLANの構成について説明します。

2.1.1 LAN構成の考慮点

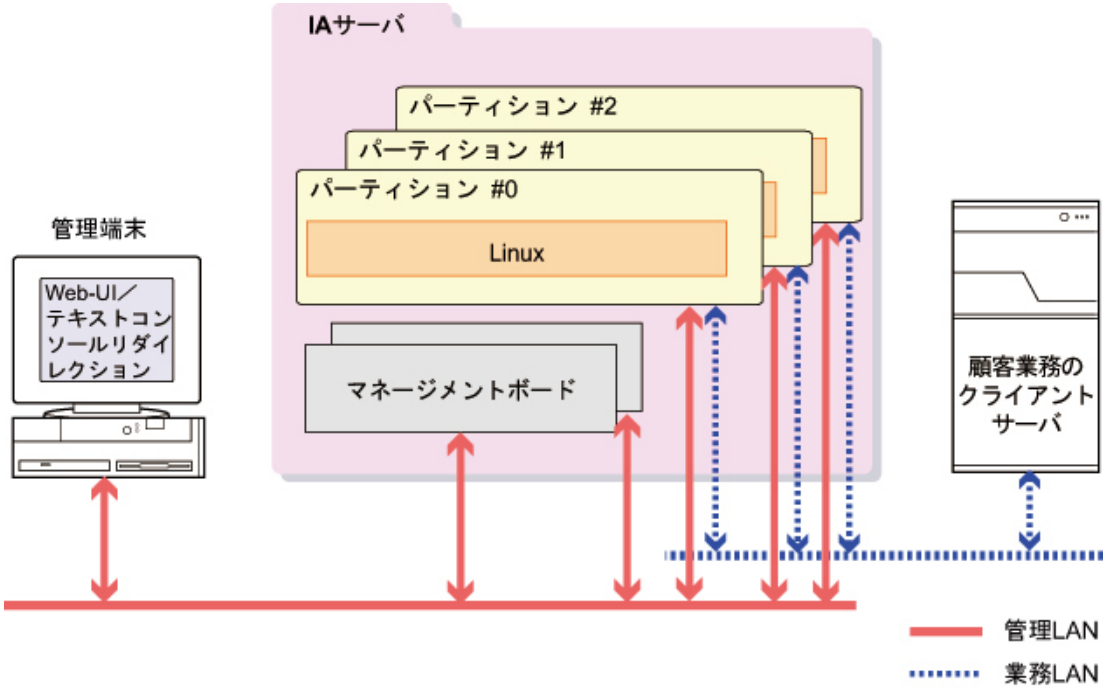
サーバが接続するLANを構成する際の考慮点について説明します。

マネージメントボードは、システム管理機能をネットワークを介して遠隔地の管理コンソールや管理ソフトウェアに提供します。このため、セキュリティと負荷分散を目的として、管理LANとは別に独立した業務LANを構成することを推奨します。

- 管理LAN
  RHELシステムを管理するために使用するLANです。
- 業務LAN
  業務システムを構築するためのLANです。

筐体内に複数のサーバ装置を収納するハードウェアでは、各パーティションを管理するための装置（マネージメントボード）が搭載されています。

図 9 : 管理装置を搭載する機種のLAN構成例



2.1.2 RHELシステムにおけるLAN接続機能

ここでは、RHELシステムをLANに接続する際に利用可能なRHELの機能について説明します。

- VLAN
  VLANは、物理的な1つの伝送路を複数の仮想的なネットワークのグループ（VLANグループ）に分割する機能です。VLANを使用したLAN環境にRHELシステムを接続する場合の設定方法、および注意事項について説明します。詳細は、「2.2 VLAN」（→P.35）をお読みください。
- bondingドライバ
  bondingは、複数のNICを組み合わせて、仮想的な1つのNICとして扱う機能です。bondingの用途と設定方法および注意事項について説明します。詳細は、「2.3 bonding」（→P.40）をお読みください。

2.1.3 ネットワークデバイスの設定・管理

サーバをLAN接続するために必要な、RHELシステムのネットワーク設定について説明します。

■ ネットワークデバイス

RHELは、サーバに搭載されたNICやbondingデバイスなど、ネットワーク通信するための操作対象をネットワークデバイスとして管理します。
ネットワークデバイスには、次表に示すデバイス名が使用されます。

表 13：ネットワークデバイス名の例

ネットワークデバイスの種類	デバイス名	デバイス名の割当て方法
NIC（LANポート単位）	eth0、eth1など	自動
bondingデバイス	bond0、bond1など	手動
VLANデバイス	eth0.1、eth0.2、eth1.1など	手動

物理的なネットワーク装置（NIC、オンボードLANなど）に対しては、RHELシステムが装置を認識した段階で自動的にデバイス名を割り当てます。しかし、bondingデバイスやVLANデバイスなど、仮想的なネットワークデバイスの場合、デバイス名は手動で割り当てます。

RHELシステムはネットワークデバイスに対して、デバイス名に対応した設定ファイルを通じて、IPアドレスの付与、特殊なネットワークの構成（VLAN、bonding）、RHELシステム起動時の活性・非活性などの設定・管理を行います。

■ ネットワークデバイス設定ファイル

ネットワークデバイスを設定するための情報（IPアドレス、DHCP、VLAN構成、bonding構成など）は、ネットワークデバイス設定ファイルに記述されます。
個々のネットワークデバイスに対する設定ファイルは、次のファイル名で配置します。

```
/etc/sysconfig/network-scripts/ifcfg-＜ネットワークデバイス名＞
```

ネットワークデバイス設定ファイルに記述する項目の詳細については、Red Hat社のマニュアル『Deployment Guide』をお読みください。

- サーバに搭載されたNIC（LANポート単位）に対するネットワークデバイス設定ファイルは、RHELインストール時に自動設定されます。しかし、RHELインストール後に増設したNICの場合は、自動設定されませんので、手動で設定します。

■ ネットワークデバイスの管理方式

ネットワークデバイス設定ファイルに基づいてネットワークデバイスを設定および起動するために、RHELは2つの管理方式を用意しています。

- networkサービスによる管理

  RHELシステム起動時にnetworkサービススクリプトが自動実行され、主にifup、ifdown、ifenslaveなどのコマンドを使用して、ネットワークデバイスの設定および起動を行います。

- NetworkManagerサービスによる管理

  RHELシステム起動時にNetworkManagerサービスによって常駐起動されるNetworkManagerデーモンが、ネットワークデバイスを管理します。主にモバイル環境におけるネットワークの動的構成変更を行うために使用されます。RHEL6.4からは、bondingデバイスおよびVLANデバイスの一部を管理できるようになりました。

ネットワークデバイスの管理方式は、ネットワークデバイス設定ファイルのNM_CONTROLLED行で設定します。NetworkManagerサービスによる管理を行う場合は「yes」を、行わない場合は「no」を設定します。

例：NetworkManager管理を行わない場合のネットワークデバイス設定ファイルの記述

```
DEVICE=eth0
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=static
IPADDR=＜IPアドレス＞
  .
  .
```

- networkサービスとNetworkManagerサービスが起動される場合、NetworkManagerサービスがネットワークデバイス設定ファイルに「NM_CONTROLLED=yes」の記述があるネットワークデバイスを管理します。

■ NetworkManagerサービスの停止

NetworkManagerサービスを停止し、networkサービスを使用したネットワークデバイスの設定および管理を推奨します。

推奨の理由は次のとおりです。

- NetworkManagerサービスは、bondingデバイスおよびVLANデバイスを完全に管理できないこと。
- NetworkManagerサービスは、クライアント環境向けのサービスであり、サーバ環境では必要ないこと。

NetworkManagerサービスの停止は、次の手順で行います。

1. すべてのネットワークデバイス設定ファイルに対して、NetworkManagerサービスによる管理をしないように設定します。

a) ネットワークデバイス設定ファイルを開きます。

例：ネットワークデバイス名がeth0の場合

```
# vi /etc/sysconfig/network-scripts/ifcfg-eth0
```

b) NM_CONTROLLED行が存在する場合、設定値を「no」に設定して、ネットワークデバイス設定ファイルを保存します。

```
DEVICE=eth0
NM_CONTROLLED=no
ONBOOT=yes
BOOTPROTO=static
IPADDR=x.x.x.x
  .
  .
```

- NM_CONTROLLED行が存在しない場合は、NM_CONTROLLED行を追加します。

2. 現在稼働中のNetworkManagerデーモンを停止し、NetworkManagerサービスの自動起動を抑止する設定およびnetworkサービスの自動起動設定を確認します。

a) NetworkManagerサービスを停止します。次のコマンドを実行し、NetworkManagerサービスの稼働状況を確認します。

```
# service NetworkManager status
```

次の出力結果の場合、NetworkManagerサービスが稼働していません。手順bに進んでください。

```
NetworkManager is stopped
```

次の出力結果の場合、NetworkManagerサービスが稼働中です。

```
NetworkManager (pid ＜PID＞) is running...
```

この場合、次のコマンドを実行し、稼働しているNetworkManagerサービスを停止します。

```
# service NetworkManager stop
```

次の出力結果の場合、NetworkManagerサービスは停止しています。

```
Stopping NetworkManager daemon:        [  OK  ]
```

b) RHELシステム起動時のNetworkManagerサービスの起動設定状況を確認するため、次のコマンドを実行します。

```
# chkconfig --list NetworkManager
```

次に示すのは、出力の一例です。

```
NetworkManager   0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

ここで、0～6までの数字は、RHELシステム起動時のランレベルを意味します。on/offは、そのランレベルでシステムが起動された場合のサービス自動起動の有無を表します。サービスの起動が「on」になっているランレベルを指定して、次のコマンドを実行します。

```
# chkconfig --level ＜ランレベル＞ NetworkManager off
```

c) RHELシステム起動時のnetworkサービスの起動設定状況を確認するため、次のコマンドを実行します。

```
# chkconfig --list network
```

次は、出力の一例です。

```
network    0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

RHELシステム起動時のデフォルトのランレベルにおけるサービスの起動が「off」になっている場合、次のコマンドを実行してサービスを「on」に設定します。

```
# chkconfig --level ＜ランレベル＞ network on
```

2.2 VLAN

VLANネットワークを構築する場合の設定方法および注意事項について説明します。

2.2.1 VLANとは

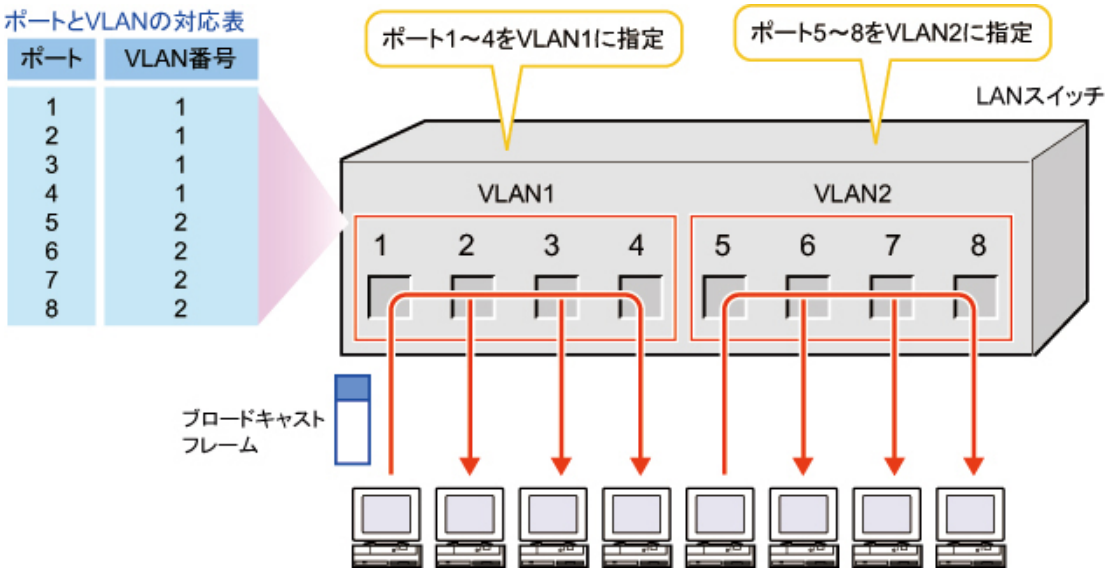
VLAN は、物理的な1つの伝送路を、複数の仮想的なネットワークのグループ（VLANグループ）に分割する機能です。同じ伝送路上の機器同士でも、VLANグループが異なると通信ができません。
VLANをサポートするスイッチは、portVLANと、tagVLANの2種類のデバイスを提供します。

■ portVLAN

portVLANは、LANスイッチのポートごとにVLAN番号を設定するVLANの方式で、スイッチ内で同じVLAN番号を持つポートが、1つのセグメントとして定義されます。これにより、スイッチ内を論理的に分割することができます。

RHELシステムをportVLANに接続する場合、特別な設定は不要です。

図 10 : portVLANの概要



■ tagVLAN

tagVLANは、ネットワークを流れるフレームにVLANタグと呼ばれるVLAN識別情報を付加することで、複数のセグメントを論理的に構築するVLANの方式です。VLANタグにはVLAN番号が設定され、これによりVLANの識別が行われます。tagVLANは、1つのポートを複数のVLANに所属させる場合に使用します。

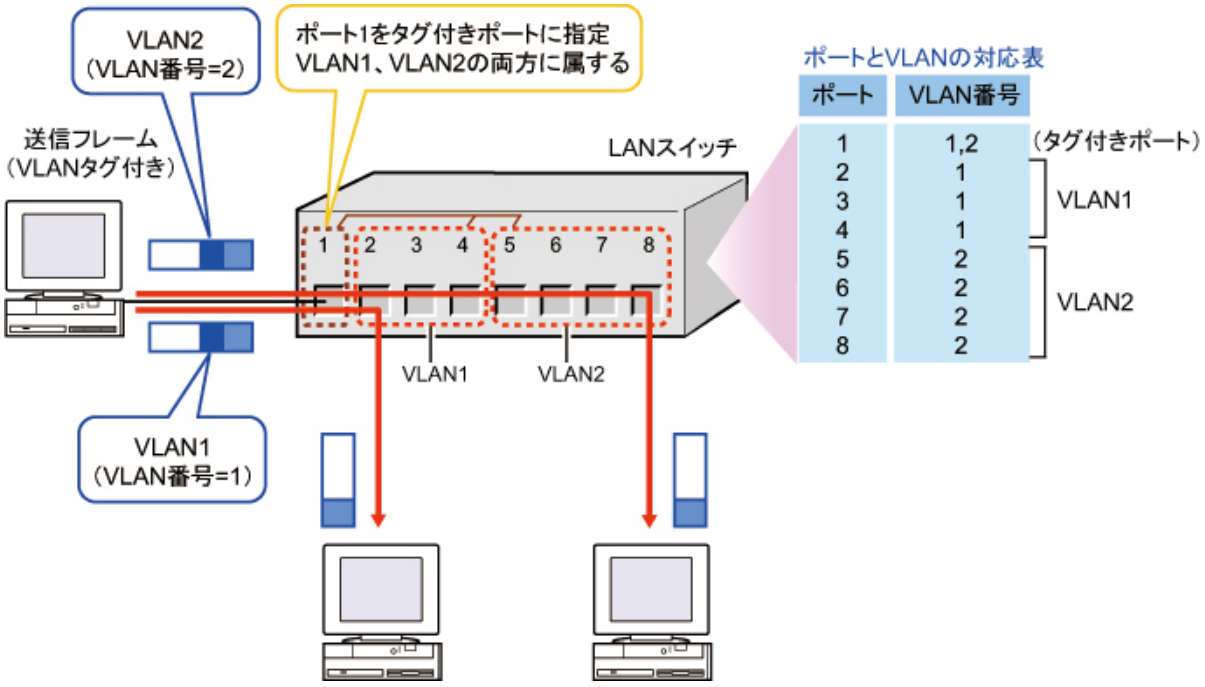
tagVLANを構築するには、スイッチやサーバのポートがVLANタグ付きフレームの送受信を可能とするための、タグ付きポートとしての設定が必要になります。

tagVLANは、主に次の用途で使用します。

- 複数のスイッチに及ぶVLANを構成する場合のスイッチ間の接続
  複数のスイッチに及ぶVLANを構成する場合、スイッチ間の接続をtagVLANで構成することにより、物理的な接続を1つにすることができます。
- サーバを1つのNICで複数のネットワークに接続する場合
  ネットワークやネットワークに属する機器を監視するなど、業務上の目的でサーバ機を複数のネットワークセグメントに所属させる必要がある場合、サーバがtagVLANに接続可能であれば、1つのNICで複数のネットワークセグメントに通信できるようになります。

RHELシステムは、VLANドライバを使用することでVLANタグ付きフレームを送受信することが可能となり、tagVLANへの接続が可能となります。設定方法の詳細は、「2.2.2 VLANの設定」（→P.36）をお読みください。

図 11 : tagVLANの概要



2.2.2 VLANの設定

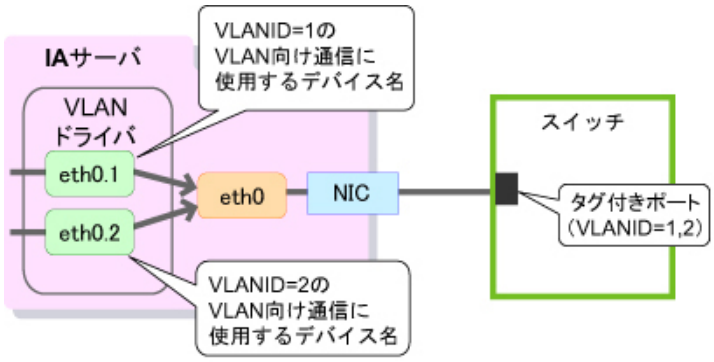
tagVLANを使用する場合、tagVLANをサポートするスイッチに接続されているネットワークデバイス上に、仮想的なネットワークデバイスをVLAN用に作成します。

仮想的なネットワークデバイスのデバイス名は、次のルールで決定されます。

```
＜ネットワークデバイス名＞.＜VLANID＞
```

例えば、eth0デバイスを経由して「VLANID=2」のtagVLANに接続する場合は、eth0.2というデバイス名に対するネットワークデバイス設定ファイルを作成します。

図 **12 : tagVLAN** 用のネットワークデバイスの概念図



注 意

- bondingと組み合わせる場合の注意事項については、「2.3.4 bondingドライバとVLANの組合せ」（→P.45）をお読みください。

■ ネットワークデバイス設定ファイルの記述方法

ネットワークデバイス設定ファイルの記述方法について説明します。

注 意

- 従来互換のために、/etc/sysconfig/networkファイルに「VLAN=yes」の行を追加する方法が残されています。しかし、この方法ではkdumpが誤動作する場合がありますので、使用しないでください。

tagVLANに相当する構成ファイルを追加します。

参 考

- tagVLANに相当する構成ファイル名の形式は、「ifcfg-＜xxxx＞.＜n＞」です。＜xxxx＞はtagVLANを載せるデバイス名、＜n＞はVLANIDです。

例：eth1にVLANID「2」を割り当てた場合

/etc/sysconfig/network-scripts/ifcfg-eth1ファイルを基にして、/etc/sysconfig/network-scripts/ifcfg-eth1.2ファイルを作成し、構成情報を設定します。

```
DEVICE=eth1.2
BOOTPROTO=static
IPADDR=＜IPアドレス＞
ONBOOT=yes
VLAN=yes
  .
  .
```

例：bond0にVLANID「2」を割り当てた場合

/etc/sysconfig/network-scripts/ifcfg-bond0ファイルを基にして、/etc/sysconfig/network-scripts/ifcfg-bond0.2ファイルを作成し、構成情報を設定します。

```
DEVICE=bond0.2
BOOTPROTO=static
IPADDR=＜IPアドレス＞
ONBOOT=yes
VLAN=yes
 .
 .
```

2.2.3 VLAN使用時の注意事項

VLANを使用する場合の注意事項について説明します。

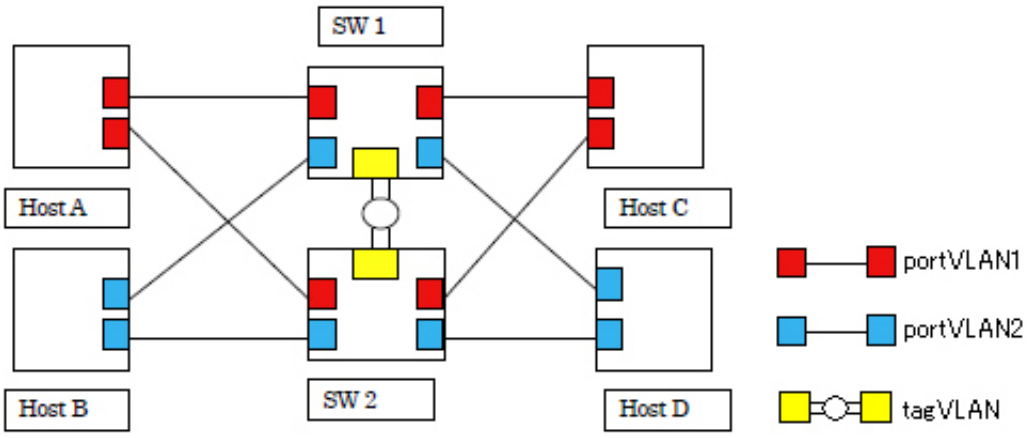
■ STPの設定

tagVLANおよびportVLANを使用してSTPを正しく動作させるには、拡張されたSTP（MSTP）が必要です。VLANを導入し、冗長化したスイッチネットワークが必要な場合には、MSTPをサポートしているスイッチを導入してください。

STPが不要な例

次図で、Host A、Host CとHost B、Host Dは独立したネットワークです。スイッチ1とスイッチ2をportVLAN（赤、青）で分割して使用しています。スイッチ間の渡りはtagVLAN（黄色）でトランキングしています。閉路がないためSTPは不要です。

図 **13 : STP**が不要なネットワーク



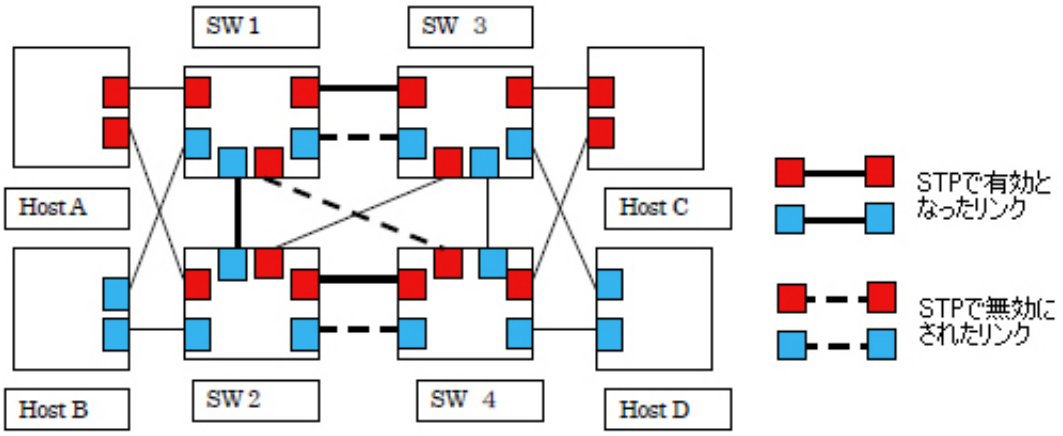
MSTPが必要な例

次図はSTPが不要な例にスイッチを1ペア追加し、VLANごとに異なる冗長路を作成した例です。閉路があるのでSTPは必須です。太線は、STPのみ（MSTPでない）の場合に有効とされるリンクの例で、どちらのVLANにも到達できないスイッチが出現します。

MSTPでは、VLANを意識して太破線も有効リンクに追加されるので、到達できないスイッチは出現しません。

実際に、どのリンクが選ばれるかは、冗長性を保証する範囲で、タイミングなどにより決まります。次図の接続形態であれば、図示したリンクが必ず選ばれるとは限りません。

図 **14 : MSTP**が必要なネットワーク



■ スイッチのIPアドレス

スイッチにはIPアドレスを付与できますが、通常、スイッチを複数のVLANで分割すると、その中の1つのVLANだけからスイッチのIPアドレスへアクセスできます。

例えば、PRIMECLUSTER GLSを導入すると、スイッチに監視用のIPアドレスが必要となります。このとき、portVLANの設定により、スイッチを独立した複数の区画に分割して、それぞれの区画をPRIMECLUSTER GLSに独立したスイッチとして認識させようとすると、片方の区画には監視用のIPアドレスが存在しないことになり、導入ができません。

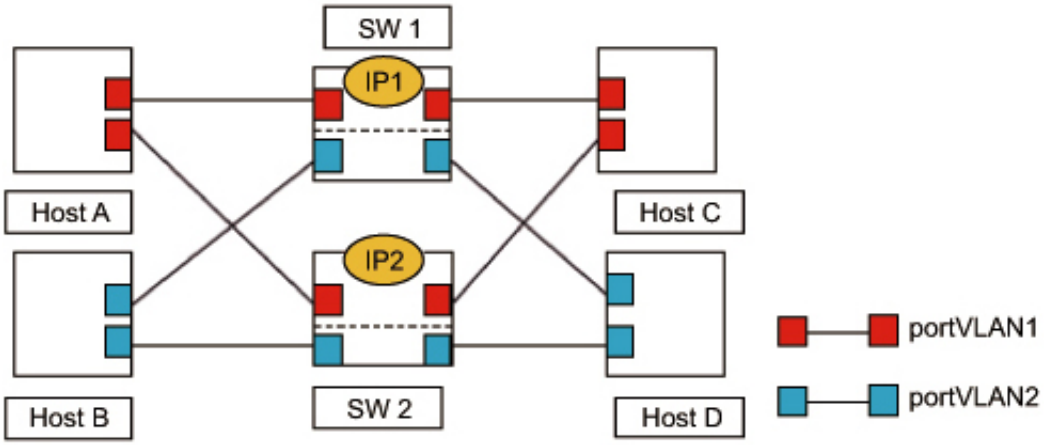
スイッチのIPアドレスを監視用途に使用する機能を導入する場合は、このことを考慮して設計してください。

または、複数のVLANで分割しても、すべてのVLANからスイッチのIPアドレスにアクセスできる機能（VLANごとにIPアドレスを割り当てることができる機能）を持ったスイッチを使用してください。

IPとVLANの関係

次図はスイッチ1、スイッチ2ともportVLAN（赤、青）で分割して使用しています。IP1、IP2とも赤のVLANのポートに割り当てられていて、青のポートからはアクセスできません。

図 **15 : IP**と**VLAN**の関係



2.3 bonding

bondingは、複数のNICを組み合わせて、仮想的な1つのNICとして扱う機能です。ここでは、bondingドライバを使用した2重化機能およびロードバランシング機能の設定方法と注意事項について説明します。

参考

- bondingの設定については、Red Hat社のマニュアル『Deployment Guide』をお読みください。

2.3.1 動作モードと監視モード

bondingドライバには7種類の動作モードと2種類の監視モードがあります。

■ 動作モード

bondingドライバには7種類の動作モードがあり、それぞれ使用目的に応じて使い分けます。

表 14 : bondingドライバの動作モード

動作モード	動作の特徴
mode0 (balance-rr)	ロードバランシング機能 スイッチのトランキングのサポートが必要 データの分散方式はラウンドロビン
mode1 (active-backup)	2重化機能専用 現用－待機構成を採り、1時点でアクティブなリンクは1本のみ
mode2 (balance-xor)	ロードバランシング機能 スイッチのトランキングのサポートが必要 データの分散方式は相手によるリンク固定
mode3 (broadcast)	ブロードキャスト用（本書では説明の対象外）
mode4 (802.3ad)	ロードバランシング機能 スイッチのトランキングのサポートが必要 データの分散方式は、IEEE802.3ad準拠
mode5 (balance-tlb)	ロードバランシング機能 スイッチのサポートが不要な方式 データの分散方式は、負荷分散方式
mode6 (balance-alb)	ロードバランシング機能 スイッチのサポートが不要な方式 データの分散方式は、mode5と同じだが、受信も分散可 IPv4のみ適用可能

動作モードを送信方式、受信方式、制限の観点で分類し、次表に記します。

送信方式は、送信データをリンクに割り当てる方式です。受信方式は、受信データをリンクに割り当てる方式です。

表 15 : 送信方式

動作モード	動作の特徴
mode0	ラウンドロビン
mode2	相手単位に固定
mode4	複数の情報をもとに固定
mode5	動的に負荷分散
mode6	

「相手単位に固定」の方式は、1つの通信相手だけに限定すると、帯域を2リンク以上に増やせないことを意味します（mode4、mode5、mode6もこの点は同様です）。

表 16 : 受信方式

動作モード	動作の特徴
mode0	スイッチの動作による
mode2	
mode4	複数の情報をもとに固定
mode5	全体で1リンクに固定
mode6	相手単位に固定

mode0、mode2の場合、動作の選択はスイッチに任されます。スイッチに設定がない場合、送信と受信で動作方式が非対称になる可能性があります。通常、スイッチはmode2相当の動作をします。なお、複数の物理スイッチを1台の論理スイッチとして構成する機能を持つ高機能スイッチの場合、異なる物理スイッチにリンクを属させることができます。

表 17 : 適用形態の制限

動作モード	適用形態の特徴
mode0	全リンクを1台のスイッチに収納する必要あり
mode2	
mode4	
mode5	複数のスイッチに及んだ動作が可能
mode6	

■ 監視モード

bondingドライバには、NICが通信可能な状態であるかを監視する監視モードが2種類あります。

監視の有無および監視モードの指定は、bondingドライバのオプションで設定します。

mii監視モード

NICとネットワーク機器とのリンク状態を直接監視する方式です。ケーブル抜けやネットワークスイッチの故障などを直接検出します。

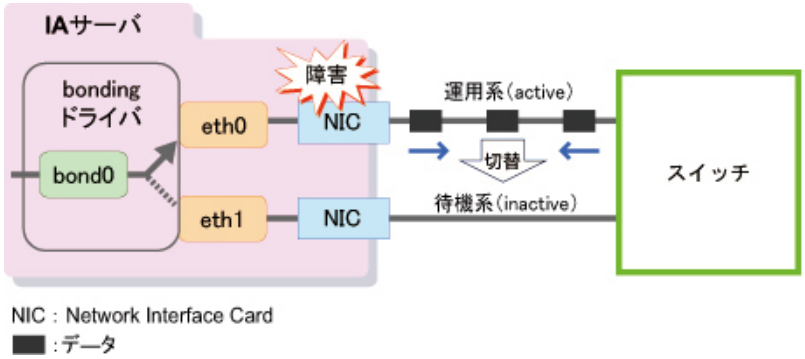
arp監視モード

arp要求パケットをネットワークに送出し、それに対してネットワークからのフレーム受信の有無を監視する方式です。何らかのフレーム受信が検出できれば、通信可能な状態と判断します。

2.3.2 2重化機能

ネットワークを2重化し、一方のネットワークをバックアップ用として待機させておきます。運用中のネットワークで障害（NIC故障、スイッチ故障、ケーブル断線など）が発生したときに、バックアップ用のネットワークに切り替えることで、ネットワークの停止を防止します。

図 16 : 2重化機能の概要



ネットワークを2重化する場合、次のモードを使用します。

- 動作モード
  必ずmode1を使用します。
- 監視モード
  mii監視モードを推奨します。

参考

- arp監視モードは、STPを導入した環境では、mii監視よりも監視できる範囲が狭くなり、適用できる形態も制限されるため、より制限のないmii監視モードを推奨します。

2重化機能を使用するためのbondingデバイスの構成方法を説明します。

1. /etc/modprobe.dディレクトリ配下からbondingの設定が記述されているファイルを特定します。
   例：/etc/modprobe.d/bonding.confファイルに記述されている場合

```
# grep -l bonding /etc/modprobe.d/*
/etc/modprobe.d/bonding.conf
```

   ファイルが存在しなかった場合は、/etc/modprobe.dディレクトリ配下に、「.conf」を拡張子とする任意のファイル名で設定ファイルを作成します（例：/etc/modprobe.d/bonding.conf）。

注意

- ドライバの設定ファイルは、必ず/etc/modprobe.dディレクトリ配下に作成してください。

2. 特定した設定ファイルにbondingデバイスの記述を追加します。

次の1行を追加します（デバイス名がbond0の場合）。

```
.
.
alias bond0 bonding
```

3. bondingデバイス用のネットワークデバイス設定ファイルを作成します。

デバイス名がbond0の場合は、ファイル名を/etc/sysconfig/network-scripts/ifcfg-bond0として、次の内容のファイルを作成します。

```
DEVICE=bond0
BOOTPROTO=static
IPADDR=＜IPアドレス＞
ONBOOT=yes
BONDING_OPTS="mode=1 primary=eth0 miimon=100"
```

bondingデバイス（masterデバイス）の設定ファイルは、通常のNIC（ifcfg-ethX）の設定に加え、BONDING_OPTS文でbondingオプションを記述します。ただし、物理MACアドレスを持たないので、HWADDR文は不要です。

BONDING_OPTS文に記述可能なオペランドは、次表のとおりです。

表 18 : 記述可能なオペランド

オペランド	説明
mode	bondingドライバのモードを指定します。「1」固定です。
miimon	リンクの状態確認の間隔をミリ秒単位で指定します。省略すると監視が行われなくなるため、必ず指定します。特に要件がなければ「100」とします。
primary	bondingデバイスを構成するスレーブデバイスのうちの1つを優先して使用したい場合に、そのデバイス名を指定します。省略しても構いません。
updelay	リンクを有効にするまでの待ち時間を、miimonパラメーターの倍数（ミリ秒）で指定します。スイッチが障害から回復したとき、リンクが不安定な状態にあると、NICの切替え通知がスイッチに到達せず、結果としてNIC切替えに時間がかかる場合があります。この状態のとき、RHELシステムからNICの切替え通知を送信しないようにすることで、短時間でNICを切り替えることができます。リンクが不安定になる場合以外、本設定は不要です。

4. bondingデバイスのslaveデバイスとして構成するすべてのネットワークデバイスに対して、ネットワークデバイス設定ファイルを修正します。

例：eth2をslaveデバイスとする場合

/etc/sysconfig/network-scripts/ifcfg-eth2ファイルを修正します。

```
DEVICE=eth2
MASTER=bond0
SLAVE=yes
ONBOOT=yes
BOOTPROTO=static
```

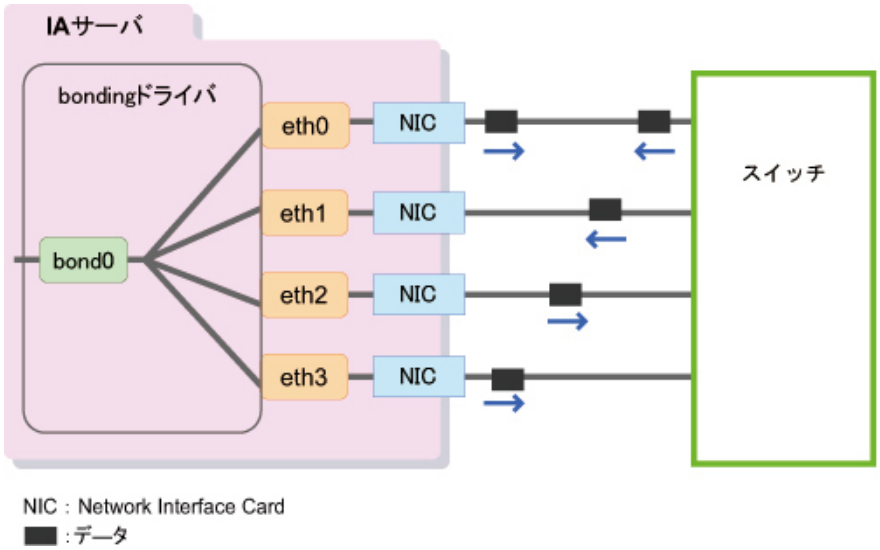
参考

- VLANと組み合わせる場合は、ifcfg-bond0にIPアドレスを設定しないで、ifcfg-bond0.XのVLAN用の設定ファイルを追加します。

2.3.3 ロードバランシング機能

ロードバランシング機能は、複数のNICをグループ化して帯域を拡張することで、ネットワークの負荷分散を実現する機能です。

図 **17 :** ロードバランシング機能の概要



bondingドライバを使用したロードバランシングを構築する場合、次のモードを推奨します。

- 動作モード
  増設LANを必要な枚数（達成したい能力による）導入し、外部スイッチとの間をmode4（IEEE802.3ad）で接続します。
- 監視モード
  mii監視モードを設定します。無応答による異常検出の間隔のデフォルト値は、30秒です。間隔を短くしたい場合は、「lacp_rate=1」を指定すると、異常検出の間隔が1秒になります。

参考

- mode4は、mode0、mode2、mode5、およびmode6を事実上包含していること、かつ最新の方式であることから、送信、受信方式ともにmode4を推奨します。
  mode5およびmode6はスイッチのサポートを必要とせず、かつリンクを複数のスイッチに分散させたい（2重化を同時に実現したい）場合の選択肢です。この場合、機能的に上位のmode6を使用します。

ロードバランシング機能を使用するためのbondingデバイスの構成方法を説明します。

1. /etc/modprobe.dディレクトリ配下からbondingの設定が記述されているファイルを特定します。

例：/etc/modprobe.d/bonding.confファイルに記述されている場合

```
# grep -l bonding /etc/modprobe.d/*
/etc/modprobe.d/bonding.conf
```

ファイルが存在しなかった場合は、/etc/modprobe.dディレクトリ配下に、「.conf」を拡張子とする任意のファイル名で設定ファイルを作成します（例：/etc/modprobe.d/bonding.conf）。

- ドライバの設定ファイルは、必ず/etc/modprobe.dディレクトリ配下に作成してください。

2. 特定した設定ファイルにbondingデバイスの記述を追加します。

次の1行を追加します（デバイス名がbond0の場合）。

```
.
.
alias bond0 bonding
```

3. bondingデバイス用のネットワークデバイス設定ファイルを作成します。

デバイス名がbond0の場合は、ファイル名を/etc/sysconfig/network-scripts/ifcfg-bond0として、次の内容のファイルを作成します。

```
DEVICE=bond0
BOOTPROTO=static
IPADDR=＜IPアドレス＞
ONBOOT=yes
BONDING_OPTS="mode=4 miimon=100"
```

bondingデバイス（masterデバイス）の設定ファイルは、通常のNIC（ifcfg-ethX）の設定と変わりません。ただし、物理MACアドレスを持たないのでHWADDR文は不要です。

4. bondingデバイスのslaveデバイスとして構成するすべてのネットワークデバイスに対して、ネットワークデバイス設定ファイルを修正します。

例：eth2をslaveデバイスとする場合

/etc/sysconfig/network-scripts/ifcfg-eth2ファイルを修正します。

```
DEVICE=eth2
MASTER=bond0
SLAVE=yes
ONBOOT=yes
BOOTPROTO=static
```

- VLANと組み合わせる場合は、ifcfg-bond0にはIPアドレスを設定せずに、ifcfg-bond0.XのVLAN用の設定ファイルを追加します。

2.3.4 bondingドライバとVLANの組合せ

bondingドライバを使用したネットワークの設計は、動作モード、監視モードまたはVLAN接続の組合せで、様々なパターンが考えられます。

■ 利用目的と推奨する組合せ

利用目的別に推奨する組合せパターンは、次表のとおりです。

表 **19 :** 利用目的に対する推奨パターン

利用目的	動作モード	監視モード	VLAN接続
2重化機能	mode1	mii監視	可能
ロードバランシング機能（PRIMEQUESTシリーズ）	mode4（接続するネットワークスイッチは、IEEE802.3ad仕様に準拠していること）		

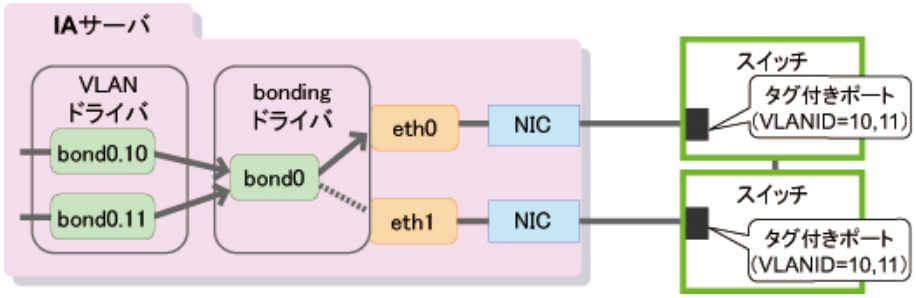
■ VLANネットワークとの組合せ時の注意事項

2重化機能/ロードバランシング機能と、VLANを組み合わせる場合の注意事項について説明します。

bondingドライバが結合するデバイスについて

bondingドライバが結合するデバイスは、任意のNICを選べますが、これらは、物理デバイスでなければなりません。例えば、bondingドライバとtagVLANデバイスが共存することは可能ですが、この場合、slaveデバイスとしてethX、ethYを必ず選択し、それらを束ねたbondZ上にbondZ.10などを作成してください。slaveデバイスとしてethX.10、ethY.10などの論理デバイス（VLANデバイス）を選択する構成はサポートされません。

図 **18 : VLAN**と**bonding**の組合せ例



mii監視モードと**VLAN**

mii監視モードで**VLAN**を設定しない場合

特に注意事項はありません。portVLANを使用する場合は、サーバにはVLANの設定は不要ですが、リンクを収納するスイッチのポートがportVLANとして構成される場合は、ネットワーク全体では、VLAN設定ありとして設計してください。

mii監視モードで**VLAN**を設定する場合

mii監視モードでVLANを設定する場合は、VLANを設定したスイッチを冗長化し、MSTPプロトコルをサポートするスイッチを使用して、スイッチのMSTPプロトコル機能を有効にしてください。

2.3.5 コマンドによる**bonding**デバイスの操作

bondingデバイスの状態は、ifconfigコマンド、ifup/ifdownコマンド、およびifenslaveコマンドで変更します。

bondingデバイス本体と構成するネットワークインターフェースを対比して記述する場合、masterデバイス、slaveデバイスと表現します。この場合、masterデバイスは、bondingデバイスを意味します。

ifconfigコマンドとifup/ifdownコマンドには、操作の対象とするmasterデバイスまたはslaveデバイスを1つ指定します。ifenslaveコマンドは、bondingデバイスの構成および状態を変更するコマンドであり、masterデバイスおよびslaveデバイスの両方を同時に指定します（詳細は指定するオプションにより異なります）。

■ ifconfigコマンド

通常のデバイスと同様に、masterデバイスおよびslaveデバイスの活性・非活性状態の設定、IPアドレスの付与などを行います。bondingデバイスの構成を変更する場合、後述のifup/ifdownコマンドまたはifenslaveコマンドを使用します。ifconfigコマンドの詳細および使用例は、オンラインマニュアルを参照してください。

■ ifup/ifdownコマンド

bondingデバイス全体（masterデバイス）とslaveデバイスの活性・非活性状態を変更します。masterデバイスとslaveデバイスでは、コマンドの動作が異なります。また、ifconfigコマンドと異なり、対応する設定ファイル（ifcfg-bondX、ifcfg-ethX）が必要です。

masterデバイスを対象にした場合

ifupコマンドを実行すると、ifcfg-ethXファイル群を検索して、対象のmasterデバイスに属するすべてのslaveデバイスを組み込んで活性状態にします。さらに、masterデバイスを活性状態にし、必要に応じてIPアドレスの割当てなどをifcfg-bondXファイルの記述に従って行います。

ifdownコマンドを実行すると、masterデバイスに属するすべてのslaveデバイスを取り外して非活性状態にします。さらにmasterデバイスも非活性状態にします。

slaveデバイスを対象にした場合

ifupコマンドを実行すると、対象となるifcfg-ethXファイルからmasterデバイスを見つけ、slaveデバイスとして組み込んで活性状態にします。

ifdownコマンドを実行すると、slaveデバイスをmasterデバイスから取り外して非活性状態にします。ifup/ifdownコマンドの詳細および使用例は、オンラインマニュアルを参照してください。

■ ifenslaveコマンド

masterデバイスへのslaveデバイスの組込みおよび取外し、アクティブなslaveデバイスの変更などを行います。主に、一時的に構成を変更するために使用します。詳細は、オンラインマニュアルを参照してください。次に使用例を示します。

masterデバイス「bond0」にslaveデバイス「eth2」を組み込む例

```
# ifenslave bond0 eth2
```

masterデバイス「bond0」からslaveデバイス「eth2」を取り外す例

```
# ifenslave -d bond0 eth2
```

masterデバイス「bond0」でアクティブ状態となっているスレーブインターフェースを「eth1」に切り替える例

```
# ifenslave -c bond0 eth1
```

ifenslaveコマンドは、masterデバイスおよびslaveデバイスを直接指定できるので、ifcfg-ethXファイルの内容にかかわらず、任意の構成を作成できます。また、ifenslaveコマンドの結果は、RHELシステムを再起動すると無効となるため、恒久的な変更の場合は、ifcfg-bondXおよびifcfg-ethXファイルを追加または変更します。

3 システム設計：デバイス名ずれを考慮した設計

ここでは、デバイス名ずれの対処方法について説明します。

3.1 デバイス名とデバイス名ずれ

デバイス名とデバイス名ずれについて説明します。

■ デバイス名

RHELは、ディスク、ネットワークなどにアクセスする場合、デバイスファイルを使用します。デバイスファイルは、デバイスと1対1に対応しているため、デバイスファイル名をデバイス名と呼んでいます。

ディスク系デバイスのデバイスファイルは、RHELシステム起動時に/devディレクトリ配下に作成され、RHELシステムが認識した順序で名前が付けられます。

名前の末尾には、ユニットと呼ばれるアルファベット（a,b,c,…）が付きます。また、ディスクパーティションの場合には、そのあとに番号（0,1,2,…）が付きます。

表 20 : デバイス名の例

デバイス名	対応する周辺機器	システムの認識順序
/dev/sda	SCSIデバイス	1番目に認識
/dev/sda1	SCSIデバイス（ディスクパーティション）	/dev/sdaと同じタイミングで認識
/dev/sdb	SCSIデバイス	2番目に認識
/dev/sdc	SCSIデバイス	3番目に認識

なお、ネットワーク系デバイス（LANポート単位）のデバイス名は、単に「eth＜整数＞」という形式であり、/devディレクトリ配下にはデバイスファイルが作成されません。しかし、ディスク系デバイスと同様に、RHELシステムが認識した順序でeth0、eth1のようにデバイス名が付けられます。

■ デバイス名ずれ

デバイス名ずれとは、USBデバイスの接続、ディスクパーティションの追加・変更、NICの追加・変更などの要因により、デバイスの認識順番が変更となり、同じデバイスを表すデバイス名が変更されるLinuxの仕様による動作のことです。

注 意

- デバイス名は、いろいろな要因で変更されるため、設定ファイル（アプリケーション設定、システム設定など）の中でデバイスを指定する場合、デバイス名の使用は推奨しません。対処については、「3.3 ディスク系デバイスにおける対処」（→P.50）および「3.4 ネットワーク系デバイスにおける対処」（→P.54）をお読みください。

3.2 デバイス名ずれによるトラブル

デバイス名ずれが発生することで、デバイス名が示すデバイスが変わります。これによって、様々なトラブルが想定されます。

■ デバイス名ずれが発生するデバイス

デバイス名ずれは、次の2種類のデバイスで発生します。

- ディスク系デバイス
  ディスク（ディスクドライブやコントローラなど）
- ネットワーク系デバイス
  NIC

■ 影響を受けるコマンドと設定ファイル

デバイス名を使ったコマンドや設定ファイルは、デバイス名ずれが発生すると正常に動作しない場合があります。影響を受ける主要なコマンドと設定ファイルを次表に記します。

表 21 : デバイス名ずれで影響を受ける主要なコマンドと設定ファイル

操作の目的	コマンド	設定ファイル
パーティションの設定	parted, fdisk	/etc/fstab
ブートローダの設定	grub-install	/boot/grub/grub.conf［注1］
ファイルシステムの操作	mkfs, mkfs.ext2, mkfs.ext3, e2label, fsck, e2fsck, tune2fs, mount, umount	/etc/fstab, /etc/auto.master
バックアップ・リストア	dump, restore	
ダンプ設定		/etc/kdump.conf
Logical Volume Manager（LVM）の操作	pvcreate, vgcreate	/etc/lvm/cache,/etc/lvm/lvm.conf
rawデバイスの操作［注2］	raw	/etc/udev/rules.d/60-raw.rules
swapファイルの操作	mkswap, swapon, swapoff	/etc/fstab
ネットワークデバイスの操作	ifconfig, ifup, ifdown	/etc/sysconfig/network-scripts/ifcfg-eth＜整数＞

［注1］：ファームウェアがUEFIモードの場合は、/boot/efi/EFI/redhat/grub.confになります。

［注2］：rawデバイスの使用は推奨しません。

■ ディスク系デバイスにおけるトラブル

ディスク系デバイスのデバイス名ずれにより、次のトラブルが想定されます。

- システムが起動できない
  ディスク系デバイスがルートファイルシステムとして使用される場合、デバイス名ずれが発生すると、RHELシステム起動時にルートファイルシステムがマウントされないため、RHELシステムが起動できなくなります。

- データ参照ができない
  ディスク系デバイスがファイルシステムとして使用される場合、デバイス名ずれが発生すると、意図するパーティションがマウントされないため、業務アプリケーションやミドルウェアなどがデータを参照できなくなります。
- データを破壊する
  ディスク系デバイスがファイルシステムとして使用される場合、デバイス名ずれが発生すると、意図しないパーティションがマウントされるため、業務アプリケーションやミドルウェアなどの動作によっては、データを破壊する場合があります。

■ ネットワーク系デバイスにおけるトラブル

ネットワーク系デバイスの場合、デバイス名ずれが発生すると、ネットワークデバイスの設定（IPアドレスの設定、bondingデバイス、VLANデバイスの設定）が意図したとおりに反映されないため、業務アプリケーションの動作、ミドルウェアの動作、またはネットワーク環境に影響を与える場合があります。

3.3 ディスク系デバイスにおける対処

デバイス名ずれの発生により、RHELシステムが起動できない、データ参照ができないなどのトラブルを未然に防ぐ対策が必要となります。

3.3.1 udev機能の利用

udev機能を使用したデバイス名ずれの対処方法を説明します。
udev機能は、次のデバイス識別子を作成します。

表 22 : udev機能によるデバイス名の割当て

デバイス名	本書での表記	説明
/dev/disk/by-id/yyyyyyyy	by-id名	/dev/disk/by-id/ディレクトリ配下に作成される識別子 yyyyyyyyはディスクに設定されている識別情報から生成される情報
/dev/disk/by-path/xxxxxxxx	by-path名	/dev/disk/by-path/ディレクトリ配下に作成される識別子 xxxxxxxxはディスクの位置情報から生成される情報 この機能の使用は推奨しません。ただし、iSCSIは例外です。
/dev/disk/by-uuid/zzzzzzzz	by-uuid名	/dev/disk/by-uuid/ディレクトリ配下に作成される識別子 zzzzzzzzはファイルシステムに設定されたUUIDをベースに生成される情報 この機能の使用は推奨しません。

デバイス名	本書での表記	説明
/dev/disk/by-label/wwwwwwww	by-label名	/dev/disk/by-label/ディレクトリ配下に作成される識別子 wwwwwwwwはファイルシステムに設定されているラベル名をベースに生成される情報 この機能の使用は推奨しません。
/dev/sdaなど	互換デバイス名	従来のデバイス名

- レスキューモードを利用する場合も、デバイス名ずれが発生する可能性があります。このため、レスキューモードを利用するときもudev機能を使用します。

■ by-id名

by-id名は、ディスク固有の識別情報（シリアル番号など）から生成されるデバイス識別子です。ディスク固有の識別情報を使うので、ディスクが交換されない限り、デバイス識別子は変わりません。ディスクを交換すると、ディスク固有の識別情報が変更となるので、by-id名が変わります。詳細については、「3.3.2 ディスク交換のby-id名変更」（→P.52）をお読みください。

■ by-id名の使用方法

by-id名は、互換デバイス名の代わりに使用します。

例：by-id名の使用

```
# mount -t ext3 /dev/disk/by-id/scsi-＜ディスクの固有情報＞-part1 /this_year
```

互換デバイス名からby-id名を求める方法

by-id名は、udevadmコマンドで表示します。次に、/dev/sdaのby-id名を表示する例を示します。下線部がby-id名です。

```
# udevadm info -q symlink -n /dev/sda
block/8:0 disk/by-id/scsi-＜ディスクの固有情報＞ disk/by-path/＜ディスクの位置情報から生成される情報＞ disk/by-id/wwn-＜ディスクの固有情報＞
```

by-id名から互換デバイス名を求める方法

by-id名は、対応する互換デバイス名へのシンボリックリンクのため、lsコマンドで確認できます。次に表示例を示します。

```
# ls -l /dev/disk/by-id/scsi-＜ディスクの固有情報＞
lrwxrwxrwx 1 root root 9 Jul 12 2010
/dev/disk/by-id/scsi-＜ディスクの固有情報＞ -> ../../sda
```

3.3.2 ディスク交換のby-id名変更

ディスクを交換した場合、交換したディスク固有の識別情報を使ってby-id名が生成されるため、交換後のディスクのby-id名は交換前のby-id名と異なります。

表 23 : by-id名が変更される操作

対象デバイス［注1］	操作	対処方法
• 内蔵SASディスク • SSDディスク • SANディスク	ディスクの活性交換［注2］	「ディスクの活性交換時のby-id名再設定手順」（→P.52）
	ディスクの非活性交換	「ディスクの非活性交換時のby-id名再設定手順」（→P.53）

［注1］：機種によっては、サポートされていないデバイスがあります。詳細は、『システム構成図』をお読みください。

［注2］：ディスクの活性交換が可能な条件については、『運用管理マニュアル』を参照してください。

ディスク交換を行った場合、コマンドの引数、システムの設定ファイル、ミドルウェアの設定ファイルなどに指定したby-id名の変更が必要となります。そのため、ディスクのby-id名を確認し、by-id名を置き換えてください。

■ ディスクの活性交換時のby-id名再設定手順

サーバの電源を切断せずにディスクの交換を行う場合（活性交換）の、by-id名の再設定手順を示します。

ここでは、次の条件を例として説明します。

- 交換するディスクは、SASディスクとします。
- マウントポイント/mntにマウントするファイルシステムに対応するデバイスのby-id名は、/etc/fstabファイルに記述されているものとします。

1. 交換するディスクを使用するサービスなどを停止し、ディスクをアンマウントします。

例：ディスクのアンマウント

```
# umount /mnt
```

2. ディスクを活性交換します。

ディスクの活性交換手順の詳細は、ハードウェアマニュアルをお読みください。

3. by-id名を指定している箇所を参照し、ディスク交換前のby-id名を確認します。

```
# cat /etc/fstab
/dev/disk/by-id/scsi-＜ディスクの固有情報①＞-part1 /mnt ext3 defaults 0 0
…
```

この例では、パーティション情報「-part1」を除いた「/dev/disk/by-id/scsi-＜ディスクの固有情報①＞」が変更前のディスクのby-id名に該当します。

4. /dev/disk/by-idディレクトリ配下と、手順3で確認したディスク交換前のby-id名を比較し、ディスク交換後のby-id名を確認します。

```
# ls /dev/disk/by-id/ | grep -v wwn
scsi-＜ディスクの固有情報②＞
scsi-＜ディスクの固有情報②＞-part1
scsi-＜ディスクの固有情報②＞-part2
…
```

この例では、「/dev/disk/by-id/scsi-＜ディスクの固有情報②＞」がディスク交換後のby-id名に該当します。

5. ディスクパーティションおよびファイルシステムの作成後、バックアップからデータを復元します。

6. ディスク交換前のby-id名を指定している箇所を、ディスク交換後のby-id名ですべて置き換えます。

ディスクのパーティションを指定している場合は「-part＜n＞」（＜n＞はパーティション番号）を追加してください。

7. ディスクをマウントし、手順1で停止したサービスなどを再開します。

例：ディスクのマウント

```
# mount /mnt
```

■ ディスクの非活性交換時の**by-id**名再設定手順

サーバの電源を切断してディスクの交換を行う場合（非活性交換）の、by-id名の再設定手順を示します。

ここでは、次の条件を例として説明します。

- 交換するディスクは、SASディスクを例とします。
- マウントポイント/mntにマウントするファイルシステムに対応するデバイスのby-id名は、/etc/fstabファイルに記述されているものとします。

1. ディスク交換後、ランレベル1（シングルユーザーモード）でRHELシステムを起動します。

2. by-id名を指定している箇所を参照し、ディスク交換前のby-id名を確認します。

```
# cat /etc/fstab
…
/dev/disk/by-id/scsi-＜ディスクの固有情報①＞-part1 /mnt vfat defaults 1 1
…
```

この例では、パーティション情報「-part1」を除いた「/dev/disk/by-id/scsi-＜ディスクの固有情報①＞」が変更前のディスクのby-id名に該当します。

3. /dev/disk/by-idディレクトリ配下と、手順2で確認したディスク交換前のby-id名を比較し、ディスク交換後のby-id名を確認します。

```
# ls /dev/disk/by-id/ | grep -v wwn
scsi-＜ディスクの固有情報②＞
scsi-＜ディスクの固有情報②＞-part1
scsi-＜ディスクの固有情報②＞-part2
…
```

この例では「/dev/disk/by-id/scsi-＜ディスクの固有情報②＞」がディスク交換後のby-id名に該当します。

4. パーティションおよびファイルシステムの再作成後、バックアップからデータを復元します。

5. ディスク交換前のby-id名を指定している箇所を、ディスク交換後のby-id名ですべて置き換えます。
   ディスクのパーティションを指定している場合は「-part＜n＞」（＜n＞はパーティション番号）を追加してください。

6. RHELシステムを再起動します。

```
# shutdown -r now
```

3.3.3 ミドルウェアの利用

デバイス名を管理する次のミドルウェア製品を導入することにより、デバイス名ずれの対処ができます。
機能の詳細は、製品のマニュアルをお読みください。

■ PRIMECLUSTER GDS

ディスク装置をミラーリングすることで、ディスク故障などの不測の事態からデータを守るボリューム管理ソフトウェアです。

このミドルウェアの持つ、RHELシステム構築時と異なるデバイス名でディスクがマウントされることを防ぐ（ディスク切離し）機能により、データ破壊を防止します。

3.4 ネットワーク系デバイスにおける対処

ネットワーク系デバイスの場合、デバイスごとに割り当てられている一意の情報（ハードウェアアドレス）を利用することでデバイス名ずれに対処します。

3.4.1 udev機能の利用

udev機能を使ったデバイス名ずれ防止方法を説明します。
なお、ここでの説明は、bondingを構成するslaveデバイスの場合も有効です。

■ デバイス名ずれ防止の仕組み

ネットワーク系デバイスのデバイス名ずれを防ぐ仕組みについて説明します。
NICを交換または増設してもRHELシステム再起動時にデバイス名が変わらないようにするために、ネットワーク系デバイス情報を次のファイルに記述します。

- udevルールファイルに、ハードウェアアドレスとデバイス名の対応を記述する。
- udevルールファイルに記述したデバイス名に対して、ネットワークの設定情報を記述したネットワークデバイス設定ファイルを作成する。

表 24 : ネットワーク系デバイスの設定関連ファイル

ファイル名	役割
udevルールファイル /etc/udev/rules.d/70-persistent-net.rules	NICの持つLANポートのハードウェアアドレスに対して、デバイス名を記述します。

ファイル名	役割
ネットワークデバイス設定ファイル /etc/sysconfig/network-scripts/ifcfg-eth＜整数＞	udevルールファイルに定義したデバイス名に対して、ネットワーク設定情報（IPアドレスなど）を記述します。

ネットワーク系デバイスのデバイス名は、RHELシステム起動時にudevルールファイルに基づいて設定され、かつudevルールファイルに記述されたデバイス名定義は、手動で編集しない限り、変更または削除されません。そのため、udevルールファイルとネットワークデバイス設定ファイルによって、ネットワーク系デバイスに常に同じデバイス名を割り当てることができます。

RHELシステムが新しいNICを検出した場合、RHELシステムはudevルールファイルを編集し、新しいNICに対応するデバイス名定義を追加しますが、ネットワークデバイス設定ファイルは変更しません。そのため、NICを交換または増設する際は、ネットワークデバイス設定ファイルの手動編集が必要です。

- 減設したNICまたは交換前のNICに対するデバイス定義は、udevルールファイルから自動的に削除されません。デバイス定義の削除が必要な場合、udevルールファイルを手動で編集します。

次に、設定ファイルの記述例を示します。

udevルールファイル

ハードウェアアドレスで特定したLANポートに対して、デバイス名を設定します。

- 書式

```
SUBSYSTEM=="net",ACTION=="add",DRIVERS=="?*",ATTR{address}=="ハードウェアアドレス",ATTR{type}=="1",KERNEL=="eth*",name="デバイス名"
```

- 記述例：ハードウェアアドレスで特定したLANポートのデバイス名を「eth0」にする

```
SUBSYSTEM=="net",ACTION=="add",DRIVERS=="?*",ATTR{address}=="ハードウェアアドレス",ATTR{type}=="1",KERNEL=="eth*",name="eth0"
```

- 実際の記述は、改行を入れないで1行で記述してください。

ネットワークデバイス設定ファイル

指定されたデバイス名に対するネットワーク設定情報を記述します。

- 記述例：デバイス名「eth0」のネットワーク設定情報/etc/sysconfig/network-scripts/ifcfg-eth0ファイル

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=＜ブロードキャストアドレス＞
HWADDR=＜ハードウェアアドレス＞
IPADDR=＜IPアドレス＞
NETMASK=255.255.255.0
NETWORK=192.168.1.0
ONBOOT=yes
TYPE=Ethernet
NM_CONTROLLED=no
```

注 意

- ネットワークデバイス設定ファイルは、HWADDRオプション（ハードウェアアドレス）の記述は省略できます。
- HWADDRオプションに記述したハードウェアアドレスは、RHELシステム起動時にudevルールファイルに記述されたハードウェアアドレスと比較され、一致しなければネットワークデバイスは活性状態となりません。

■ RHELインストール時の自動設定

RHELをインストールすると、接続されているすべてのNICに対応するLANポートのハードウェアアドレスとデバイス名を記述したudevルールファイルと、それぞれのネットワークデバイス設定ファイルが作成されます。そのため、RHELをインストールした後、NICを交換または増設するまでは、デバイス名ずれを防ぐ対処は不要です。

NICを交換または増設する場合、デバイス名ずれを防ぐため、udevルールファイルおよびネットワークデバイス設定ファイルの編集が必要です。このとき、NICを交換または増設する前のネットワーク系デバイスのハードウェア構成情報が必要となります。そのため、ネットワーク系デバイスのハードウェア構成情報を採取し、保管してください。 なお、bondingを使用する場合、bondingの設定をする前にネットワーク系デバイスのハードウェア構成情報を採取してください。

eth0デバイスの場合を例にして説明します。

1. LANポートに対応するデバイス名を収集します。

/sys/class/netディレクトリから、ネットワークデバイス名を収集します。

```
# cd /sys/class/net
# ls
eth0 eth1 lo
```

表示された一覧のうち、「eth＜整数＞」の形式の名前がLANポートに割り当てられたデバイス名です。この情報から、ネットワークデバイス名の一覧を作成してください。

2. LANポートとハードウェアアドレスの対応を確認します。

手順1で収集したすべてのデバイス名に対応するハードウェアアドレス（MACアドレス）を確認します。

```
# cat eth0/address
＜ハードウェアアドレス＞
```

3. LANポートとバスアドレスの対応を確認します。

手順1で収集したすべてのデバイス名に対応するLANポートのバスアドレスを確認します。

```
# ls -l eth0/device
lrwxrwxrwx 1 root root 0 Apr 9 09:17 eth0/device -> ../../../
0000:01:06.0
```

コマンド出力結果で表示されるシンボリックリンク先ファイルの最下位層のファイル名（実体はディレクトリ）がバスアドレスです。

参 考

- バスアドレスは、次の情報で構成されているデバイスを識別するためのアドレス情報です。

  [＜セグメント番号＞:]＜バス番号＞:＜スロット番号＞.＜ファンクション番号＞

セグメント番号は、機種によっては表示されないため、バス番号の一部として取り扱います。

4. LANポートの物理位置を確認します。

デバイス名に対応するLANポートの物理位置を確認します。次の例では、LEDが10秒点滅します。これにより、ネットワークデバイスの物理位置を確認します。

```
# ethtool -p eth0 10
```

注 意

- LANポートが1つのNICは、NICの位置を特定してください。複数のLANポートを持つNICの場合は、NICの位置に加えてNIC上のLANポートの位置も特定してください。
- ブレードサーバの場合は、LANポートが筐体内のスイッチに直接接続されているため、LEDによる確認ができません。ハードウェアに添付されるハードウェア管理機能を使用して、LANポートと筐体内スイッチとの接続関係を確認してください。

5. LANポートに関する情報の確認結果をまとめます。

これまで収集したLANポート情報は、次表のようにまとめて、常に最新情報となるように管理します。

表 25 : ネットワーク系デバイスのハードウェア情報（例）

作成日	**yyyy/mm/dd**		
対象システム	＜サーバ名＞		
ネットワークデバイス名	ハードウェアアドレス（**MAC**アドレス）	バスアドレス	物理位置
eth0	＜ハードウェアアドレス①＞	＜バス番号＞:＜スロット番号＞.＜ファンクション番号①＞	オンボードLAN
eth1	＜ハードウェアアドレス②＞	＜バス番号＞:＜スロット番号＞.＜ファンクション番号②＞	増設スロット1／ポート1
…	…	…	…

■ ネットワークデバイス名の変更手順

ここでは、ネットワークデバイス名を変更する手順について説明します。

1. 変更後のネットワークデバイス名体系を設計します。

「RHELインストール時の自動設定」（→P.56）で作成したネットワーク系デバイスのハードウェア情報をもとに、LANポートに割り当てる新しいデバイス名を決定します。

表 **26：eth0とeth1を交換する場合の例**

作成日	**yyyy/mm/dd**		
対象システム	＜サーバ名＞		
ネットワークデバイス名	ハードウェアアドレス（**MAC**アドレス）	バスアドレス	物理位置
eth0（→eth1に変更）	＜ハードウェアアドレス①＞	＜バス番号＞:＜スロット番号＞.＜ファンクション番号①＞	オンボードLAN
eth1（→eth0に変更）	＜ハードウェアアドレス②＞	＜バス番号＞:＜スロット番号＞.＜ファンクション番号②＞	増設スロット1／ポート1
…	…	…	…

注 意

- デバイス名が重複しないことを確認してください。

2. root権限でログインします。

3. デバイス名を変更するネットワークデバイスに対応するネットワークデバイス設定ファイル（/etc/sysconfig/network-scripts/ifcfg-eth＜整数＞）の編集準備をします。

変更前のネットワークデバイス設定ファイルのファイル名を変更して、新しいデバイス名のネットワークデバイス設定ファイルの編集用に作業用ディレクトリに移動します。

例：＜ハードウェアアドレス①＞を持つネットワークデバイスの設定ファイルを移動する場合

```
# cd /etc/sysconfig/network-scripts
# grep "＜ハードウェアアドレス①＞" ifcfg-eth*
ifcfg-eth0:HWADDR=＜ハードウェアアドレス①＞
# mkdir temp
# mv ifcfg-eth0 temp/ifcfg-eth1
```

grepコマンドを使用して、＜ハードウェアアドレス①＞が記述されたネットワークデバイス設定ファイルを検索した結果、対応するネットワークデバイス設定ファイルはifcfg-eth0であることが分かりました。デバイス名をeth0からeth1に変更するため、ifcfg-eth0ファイルをifcfg-eth1にファイルを変更して作業用ディレクトリtempに移動します。デバイス名を変更するすべてのネットワークデバイス設定ファイルについて、繰り返します。

注 意

- ネットワークデバイス設定ファイルをサブディレクトリに移動することで、RHELシステムの起動時にデバイス名変更対象のNICが活性状態になることを防ぎます。これにより、デバイス名変更途中のNICを使えないようにします。

4. udevルールファイルを編集し、デバイス名の設定を変更します。

/etc/udev/rules.d/70-persistent-net.rulesファイルを編集します。

a) 変更するデバイスのハードウェアアドレスが記載されている設定行を探します。

```
SUBSYSTEM=="net",ACTION=="add",DRIVERS=="?*",ATTR{address}=="＜ハードウェアアドレス①＞",ATTR{type}=="1",KERNEL=="eth",name="eth0"
```

b) 設定行の「name="＜デバイス名＞"」を変更するデバイス名に変更します。

例：デバイス名をeth0からeth1に変更する場合

```
SUBSYSTEM=="net",ACTION=="add",DRIVERS=="?*",ATTR{address}=="＜ハードウェアアドレス①＞",ATTR{type}=="1",KERNEL=="eth",name="eth1"
```

c) 変更が必要なすべてのデバイス名を変更した後、udevルールファイルを保存します。

ファイルの保存前に、同じデバイス名が複数定義されないことを確認します。

5. 作業用ディレクトリに移動したネットワークデバイス設定ファイルを編集し、正規のディレクトリ配下に移動します。

新しいデバイスファイル名で作業用ディレクトリに移動したネットワークデバイス設定ファイルを編集し、新しいデバイス名に変更します。

a) 編集誤りに備えて、編集前のファイルを別名で保存します。

例：新デバイス名がeth1の場合

```
# cd /etc/sysconfig/network-scripts/temp
# cp ifcfg-eth1 ifcfg-eth1.org
```

b) ファイルを編集し、DEVICE行を変更前のデバイス名から変更後のデバイス名に変更して保存します。

例：新デバイス名がeth1の場合

```
DEVICE=eth1
```

c) 編集が終わったネットワークデバイス設定ファイルを正規の位置に移動します。

例：新デバイス名がeth1の場合

```
# mv ifcfg-eth1 /etc/sysconfig/network-scripts
```

d) デバイス名を変更するすべてのネットワーク設定ファイルに対して、手順aから手順cの操作を繰り返します。

6. RHELシステムを再起動します。

```
# shutdown -r now
```

7. root権限でログインした後、ネットワークデバイスの設定状況を確認します。

ネットワークデバイス名が、意図したとおりに変更されたことを確認します。

出力例

```
# ifconfig -a
eth0 Link encap:Ethernet HWaddr ＜ハードウェアアドレス②＞
       :
eth1 Link encap:Ethernet HWaddr ＜ハードウェアアドレス①＞
       :
```

8. ネットワークデバイス名の変更に問題がないことを確認した後、ネットワークデバイス設定ファイルの作業用ディレクトリを削除します。

```
# rm -rf /etc/sysconfig/network-scripts/temp
```

9. ネットワーク系デバイスのハードウェア情報管理表を更新します。

「RHELインストール時の自動設定」（→P.56）で作成したネットワークデバイス名とハードウェア構成情報の対応表を更新します。

3.4.2 ネットワーク系デバイスの交換・増設

ネットワーク系デバイスを交換または増設する場合、udevルールファイルおよびネットワークデバイス設定ファイルを編集してデバイス名ずれに対処します。

■ NICの増設手順

NICの増設手順について説明します。

この説明は、bondingを構成するslaveデバイスの場合にも有効です。

1. ネットワークデバイス名とバスアドレスの対応一覧を準備します。

「RHELインストール時の自動設定」（→P.56）で作成したネットワークデバイス名とハードウェア構成情報の対応表を用意します。

2. NICを増設します。

サーバの電源切断後、ハードウェアマニュアルに記載されているオプションカードの取付け・取外し手順に沿ってNICを増設します。NICの増設が完了した後、サーバの電源を投入してRHELシステムを起動します。

3. root権限でログインし、追加されたネットワークデバイス名を確認します。

増設したNICのデバイス情報は、udevルールファイルに自動的に追加されます。この追加されたエントリーに存在するname="eth＜整数＞"を参照し、NIC増設以前に存在していなかったネットワークデバイス名（eth＜整数＞）を確認します。

- 新たなネットワークデバイス名は、NICの持つLANポート数に応じて追加されます。

例：新たにeth3が追加された場合

```
SUBSYSTEM=="net",ACTION=="add",DRIVERS=="?*",ATTR{address}=="＜ハードウェアアドレス＞", ATTR{type}=="1",KERNEL=="eth*",name="eth3"
```

ここで確認したネットワークデバイス名が、増設されたNICのLANポートに対応します。

4. ネットワークデバイス設定ファイルを作成します。

追加したネットワークデバイス用のネットワークデバイス設定ファイルを作成します。HWADDR文に設定するハードウェアアドレスは、udevルールファイルに記述されたハードウェアアドレスを使用します。

bondingデバイスにslaveを追加する場合は、slaveのifcfg-eth＜整数＞ファイルにbondingの設定を同時に行っても構いません。bondingデバイス自体を追加する場合は、いったんslaveデバイスを通常のデバイスとして追加したあとで設定を行います。

記述例：追加されたネットワークデバイス名がeth3の場合

```
# Intel Corporation 82545GM Gigabit Ethernet Controller
DEVICE=eth3
BOOTPROTO=static
IPADDR=＜IPアドレス＞
ONBOOT=yes
HWADDR=＜ハードウェアアドレス＞
NM_CONTROLLED=no
```

5. 新しいLANポートを活性状態にします。

例：追加されたネットワークデバイス名がeth3の場合

```
# ifdown eth3
# ifup eth3
```

シングルユーザーモードで起動した場合は、コマンドを実行する代わりにRHELシステムを再起動します。

6. ネットワーク系デバイスのハードウェア情報管理表を更新します。

「RHELインストール時の自動設定」（→P.56）で作成したネットワークデバイス名とハードウェア構成情報の対応表を更新します。

■ NICの交換手順

NIC交換は、udevルールファイルおよびネットワークデバイス設定ファイルの修正と1対1に行います。複数のNICを交換する場合も、1枚ずつ交換します。

1. ネットワークデバイス名とバスアドレスの対応一覧を準備します。

交換するNICのスロット位置の確認が必要なため、「RHELインストール時の自動設定」（→P.56）で作成したネットワークデバイス名とハードウェア構成情報の対応表を用意します。

2. 交換するNICのLANポートを確認します。

複数のLANポートを持つNICの場合、それぞれのLANポートに対応するネットワークデバイス名の対処が必要です。

- 複数のLANポートを持つNICにおいて、それぞれのLANポートに対応するネットワークデバイス名を特定する方法は、「NIC上のLANポート特定方法」（→P.64）をお読みください。

3. 交換対象となるネットワークデバイス設定ファイルを移動します。

```
# cd /etc/sysconfig/network-scripts
# mkdir temp
# mv ifcfg-eth＜整数＞ temp/ifcfg-eth＜整数＞.bak
```

- 複数のLANポートを持つNICの場合、すべてのLANポートに対応するネットワークデバイス設定ファイルを移動します。

- ネットワークデバイス設定ファイルをサブディレクトリに移動することで、RHELシステムの起動時にデバイス名変更対象のNICが活性状態になることを防ぎます。これにより、デバイス名変更途中のNICを使えないようにします。

4. udevルールファイルを編集し、交換するLANポート定義行をコメント行にするか、または削除します。

記述例：eth1定義行をコメント行とします。

```
# SUBSYSTEM=="net",ACTION=="add",DRIVERS=="?*",ATTR{address}=="＜ハードウェアアドレス＞", ATTR{type}=="1",KERNEL=="eth*",name="eth1"
```

参考

- NICのLANポートが1つの場合、udevルールファイルからLANポート定義行を削除する（コメント行化を含む）ことで、RHELシステムを再起動した際、削除したデバイス名が再割当てされることから、あらかじめをLANポート定義行を削除します。
- 複数のLANポートを持つNICの場合、udevルールファイルからLANポート定義行を削除しても、RHELシステム起動時に割り当てる順序の保証はありません。そのため、RHELシステムの再起動後に、デバイス名ずれの有無を確認します。
- ネットワークデバイスに割り当てるデバイス名に欠番（eth0,eth1,eth3,eth4など）を設ける設計とした場合、同じデバイス名が割り当てられることは期待できませんので、デバイス名に欠番が生じるような設計は避けてください。

5. サーバの電源を切断し、NICを交換します。

サーバの電源切断後、ハードウェアマニュアルに記載されているオプションカードの取付け・取外し手順を参照し、NICを交換します。NICの交換が完了した後、サーバの電源を投入してRHELシステムを起動します。

6. 移動したネットワークデバイス設定ファイルを編集します。

手順3で/etc/sysconfig/network-scripts/tempディレクトリ配下に移動したネットワークデバイス設定ファイル（ifcfg-eth＜整数＞.bak）を編集し、交換したNICのLANポートに対応するMACアドレスを設定します。NICの持つLANポート数により、次の対応を行います。

- NICのLANポートが1つの場合

  LANポートのデバイス名ずれは発生しないため、移動したネットワークデバイス設定ファイルに新しいMACアドレスを設定します。

- NICのLANポートが複数の場合

  LANポートのデバイス名ずれが発生する可能性があります。

  この場合、「ネットワークデバイスのデバイス名ずれ確認とデバイス名固定手順」（→P.63）に示す方法で、各ネットワークデバイス名に対する正しいMACアドレスを確認し、移動したネットワークデバイス設定ファイル（ifcfg-eth＜整数＞.bak）に新しいMACアドレスを設定します。

7. 編集が終わったネットワークデバイス設定ファイルを正規の位置に移動します。

```
# cd /etc/sysconfig/network-scripts
# mv temp/ifcfg-eth＜整数＞.bak ifcfg-eth＜整数＞
```

- 複数のLANポートを持つNICの場合、すべてのLANポートに対応するネットワークデバイス設定ファイルを移動します。

8. udevルールファイルを確認します。

   MACアドレスおよびデバイス名が、ネットワークデバイス設定ファイルの内容と一致していることを確認します。

9. 新しいLANポートを活性状態にします。

```
# ifdown eth＜整数＞
# ifup eth＜整数＞
```

   シングルユーザーモードで起動した場合は、コマンドを実行する代わりにRHELシステムを再起動します。

- 複数のLANポートを持つNICの場合、すべてのLANポートを活性状態にします。

10. 作成した退避ディレクトリを削除します。

```
# rm -rf /etc/sysconfig/network-scripts/temp
```

11. ネットワーク系デバイスのハードウェア情報管理表を更新します。

    「RHELインストール時の自動設定」（→P.56）で作成したネットワークデバイス名とハードウェア構成情報の対応表を更新します。

■ ネットワークデバイスのデバイス名ずれ確認とデバイス名固定手順

ここでは、デバイス名ずれ発生の有無の確認手順とデバイス名の固定手順を説明します。

複数のLANポートを持つNICまたはシステムボードを交換した場合、LANポートのデバイス名ずれが発生する場合がありますので、デバイス名ずれ発生の有無を確認します。

1. 機器交換前の最新のネットワークデバイス名とハードウェア構成情報の対応表を用意します。

   「RHELインストール時の自動設定」（→P.56）で作成したネットワークデバイス名とハードウェア構成情報の対応表を用意します。

2. 機器交換後のネットワークデバイス名とハードウェア構成情報の構成表を作成します。

   「RHELインストール時の自動設定」（→P.56）に沿って作成します。

3. 機器交換前後のハードウェア構成情報の構成表を比較し、内容が一致しないデバイスを抽出します。

4. 手順3で確認された内容が一致しないデバイスに対して、「デバイス名－バスアドレス」の対応状況を機器交換前後で比較します。

   - 交換前後で「デバイス名－バスアドレス」の対応が一致している場合

     デバイス名ずれは発生していません。この場合、機器交換後のハードウェア構成情報の構成表にあるMACアドレスをネットワークデバイス設定ファイルに使用できます。

- 交換前後で「デバイス名－バスアドレス」の対応に不一致があった場合
  デバイス名ずれが発生していますので、次の対処を実施します。

  a) 対象とする機器の交換前バスアドレスを確認します。

  b) 確認したバスアドレスを交換後のハードウェア構成情報の構成表から探します。

  c) バスアドレスに対応するMACアドレスを取得し、ネットワークデバイス設定ファイルを更新します。

  d) udevルールファイルの該当するデバイス名定義行のMACアドレスを修正します。
  交換前のLANポートに対応するデバイス名定義行がudevルールファイルに残った状態の場合、交換前のデバイス名定義行を削除してください。また、新しいMACアドレスに対応するデバイス名定義行が追加されています。新しいデバイス名定義行のデバイス名を意図したデバイス名に修正してください。

  e) RHELシステムを再起動します。

```
# shutdown -r now
```

  RHELシステムを再起動後、機器交換前と同じネットワーク構成になります。

- ネットワークデバイスに割り当てる名前に故意に欠番を設けた場合（eth0,eth1,eth3,eth4など）、欠番より大きな番号のデバイス（この例ではeth3,eth4）の交換を行うと、udevは空いたより小さな番号を使ってudevルールファイルにデバイス名定義行を追加するため、RHELシステムの再起動時にデバイス名が変更される場合があります。この場合、前述の方法でMACアドレスを確認し、デバイス名の固定手順を実施してください。また、交換時の名前に関する混乱を避けるために、デバイス名に欠番が生じるような設計はしないでください。

■ NIC上のLANポート特定方法

複数のLANポートを持つNICにおいて、LANポートを特定する方法を説明します。

1. 機器交換前の最新のネットワークデバイス名とハードウェア構成情報の対応表を用意します。

「RHELインストール時の自動設定」（→P.56）で作成したネットワークデバイス名とハードウェア構成情報の対応表を用意してください。

2. 特定したいNICのLANポートのデバイス名から、バスアドレスを確認します。

ネットワーク系デバイスのハードウェア情報管理表を検索して、該当するデバイス名を探します。

3. 特定したバスアドレスと同一のバス番号および同一のスロット番号を持つデバイス名を、ネットワーク系デバイスのハードウェア情報管理表から探します。

LANポートのバスアドレスは、次の情報の組合せで構成されます。

```
[<セグメント番号>:]<バス番号>:<スロット番号>.<ファンクション番号>
```

セグメント番号は、機種によっては表示されないため、バス番号の一部として取り扱います。

このうち、ファンクション番号だけ異なり、バス番号およびスロット番号が同一のバス番号を持つLANポートは、同一のNIC上に存在することが分かります。

3.4.3 レスキューモードでネットワーク系デバイスを使用する場合の注意

レスキューモードを使用する場合のデバイス名の扱いについて説明します。

- レスキューモードの起動方法については、「6.5 レスキューモードの起動（トラブル発生時）」（→P.117）をお読みください。

■ デバイス名を考慮したネットワーク設定

レスキューモードでは、通常運用時に設定したudevルールファイルおよびネットワークデバイス設定ファイルは使用されません。そのため、レスキュー モードの起動画面またはレスキューモード起動後にifconfigコマンドを使用して、デバイス名を考慮したネットワーク設定を実施します。

例：ifconfigコマンドによるネットワークの設定

```
# /etc/rc.d/init.d/network start
# ifconfig eth0 ＜IPアドレス＞ netmask 255.255.255.0 broadcast ＜ブロードキャストアドレス＞
```

4 システム設計：ログファイル運用

ここでは、ログファイルの運用方法について説明します。

4.1 ログファイルの運用設計

システムの保守や監視に欠かせない情報を含むログファイルは、発生した事象を確実に記録する運用が必要となります。ここでは、確実な記録を実現するログファイルの運用設計について説明します。

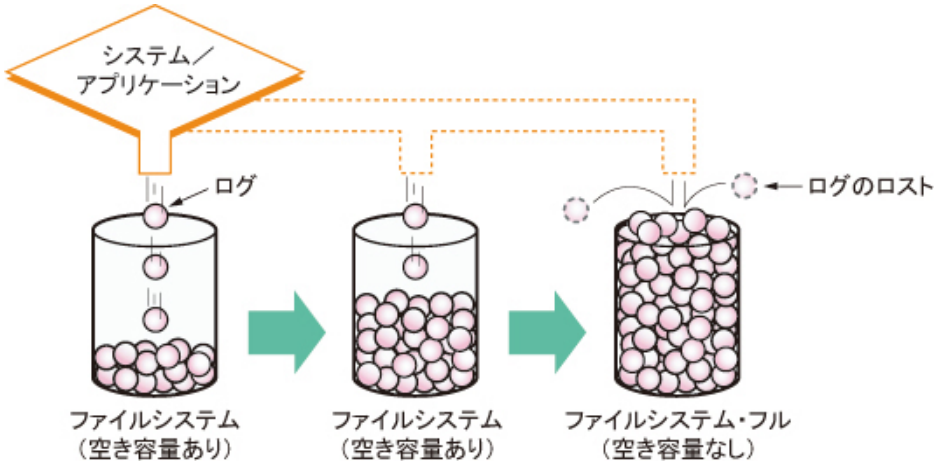
4.1.1 ログローテーション

ログファイルの運用設計をするためには、ログローテーションの設計が重要です。

■ 1つのログファイルで運用する場合の問題点

ログファイルには、OSまたはアプリケーションが書込む事象ログデータを蓄積し続けます。そのため、時間経過によりログファイルのあるディスクの空き領域が枯渇すると、それ以上は事象ログデータをログファイルに書き込めなくなります。

図 19 : 1つのログファイルで運用する場合



事象ログデータのロストを避けるためには、ディスク領域が枯渇する前に、ログファイルをバックアップ媒体に退避し、ディスク領域を解放する必要があります。

しかし、1つのログファイルだけでログファイル運用をする場合、ログファイルが常に使用中であることから、バックアップデータの内容が保証できない問題およびディスク領域を解放できない問題が発生します。

■ ログローテーション

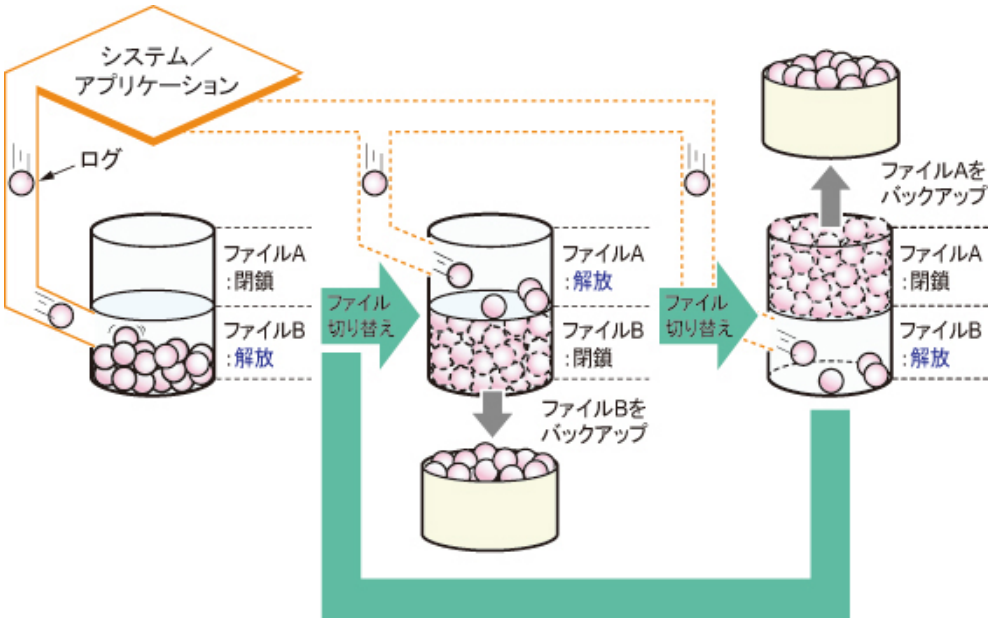
この問題を解決するために、複数のログファイルをローテーションして運用する方式があります。この運用方式をログローテーションといいます。

ログローテーションは、次のように実現されます。

1. 1つ目のログファイル（次図のファイルB）に事象ログデータを蓄積する
2. 1つ目のログファイルが切替え条件（ファイルサイズ、切替え期間など）に達すると、2つ目のログファイル（次図のファイルA）に切り替える
3. 1つ目のログファイル（次図のファイルB）をバックアップした後に、ログファイルを削除し、ディスク領域を解放する

上記1.～3.を繰り返すことで、ログファイルをローテーションします。

図 20 : ログファイルの切替え



4.1.2 ログファイルの種類

ログファイルには、システムログファイルおよびユーザー固有ログファイルの2種類があります。

■ システムログファイル

RHELが標準的に提供するシステムログファイルを次表に示します。

システムログファイルの運用設計の考え方については、「4.2 システムログファイルのログローテーション」（→P.69）をお読みください。

表 27 : システムログファイル一覧

ログファイル	ログデータ書込みプロセス	ログデータの内容	**logrotate**設定ファイルのデフォルト	
			切替え周期	保有世代数
/var/log/wtmp	各種認証プロセス（login, gdmなど）	ログインの記録（ログイン種別、プロセス番号、ユーザー名、端末名、リモートホスト名、ログイン・ログアウト時刻、終了状態など）	月ごと	1

ログファイル	ログデータ書込みプロセス	ログデータの内容	**logrotate**設定ファイルのデフォルト	
			切替え周期	保有世代数
/var/log/btmp	各種認証プロセス（login, gdmなど）	ログイン失敗の記録（ログイン種別、プロセス番号、ユーザー名、端末名、リモートホスト名、ログイン・ログアウト時刻、終了状態など）	月ごと	1
/var/log/dmesg	/etc/rc.sysinit内のdmesgコマンド	RHELシステム起動時にコンソールに出力されるメッセージの記録（CPU数、メモリ状況、各種ハードウェア状態など）	ローテーション運用なし RHELシステム起動時にデータを上書き	
/var/log/boot.log	rsyslogd	RHELシステム起動時に実行されるrcスクリプトまたは各種サービスの記録	週ごと	1［注1］
/var/log/messages	rsyslogd	RHELシステム状態の記録（デバイスドライバ、カーネル、アプリケーションにより検出された、システムの状態変化またはエラー事象が通知される）	週ごと	4
/var/log/cron	rsyslogd	クロックデーモン（cron, at）によるスケジュール実施状況の記録	週ごと	4
/var/log/secure	rsyslogd	認証関連の記録（ログインセッションの開始・終了、パスワード認証の成否など）	週ごと	4

［注1］：RHEL6.1までの保有世代数は4、RHEL6.2からの保有世代数は1です。

■ ユーザー固有ログファイル

ユーザー固有ログファイルは、システムログファイル以外のログファイルの総称です。アプリケーション、パッケージソフトウェアなどが、必要に応じて使用するログファイルです。

ユーザー固有ログファイルの運用設計の考え方については、「4.3 ユーザー固有ログファイルのログローテーション」（→P.73）をお読みください。

4.2 システムログファイルのログローテーション

システムログファイルのログローテーションについて説明します。

4.2.1 ログローテーションとスケジュール

システムログファイルのローテーションを運用するlogrotateコマンドについて説明します。

■ logrotateコマンド実行方法

logrotate コマンドは常駐プロセスではありませんので、システムログファイルを切り替えるタイミングでlogrotate コマンドを実行します。

logrotate コマンドの実行は、次の方法から選択します。

- スケジュール設定による定期的な起動（cron、Systemwalkerなどを使用）
- コマンドライン入力やスクリプトから任意のタイミングで実行

■ ログファイル切替え契機

ログファイルの切替えを検討する際、システムログファイルの負荷が高い時間帯での実施は避け、業務に影響しない時間帯に実施するよう考慮します。

スケジュールの変更方法については、「4.2.4 ログローテーションのスケジュール変更」（→P.71）をお読みください。

4.2.2 システムログファイルの切替え周期と保有世代数

システムログファイルの総量がシステム運用に適切となるよう、切替え周期および保有世代数を設計します。

切替え周期および保有世代数の設定方法については、「4.2.5 ログローテーション条件の変更」（→P.72）をお読みください。

■ システムログファイルの切替え周期

logrotateコマンドは、logrotateコマンドの設定ファイル（以降、設定ファイルと表記します）に記述されたシステムログファイルの切替え周期を確認し、切替え周期に達している場合にログファイルを切り替えます。

次表に、logrotateコマンドによるログファイル切替え周期およびログファイル切替え条件を示します。

表 28 : システムログファイル切替え周期と切替え条件

指定方法	周期指定	切替え条件
設定ファイル	monthly （月ごと）	logrotateコマンド実行日が、前回のログファイル切替え日と比べて、月が変わっている場合。 • 切り替わる例 前回の切替え日が8月30日で、コマンド実行日が9月1日の場合。

指定方法	周期指定	切替え条件
	weekly （週ごと）	logrotateコマンド実行日が、前回のログファイル切替え日と比べて、週が変わっている場合（週の始まりは、日曜日とする）。 または、前回の切替え日から7日以上経過している場合。 • 切り替わる例 前回の切替え日が土曜日で、コマンド実行日が翌日の日曜日の場合。
	daily （日ごと）	logrotateコマンド実行日が、前回のログファイル切替え日と異なる場合。 • 切り替わる例 前回の切替え日が9月1日23:59で、コマンド実行日が9月2日0:01の場合。
	size=＜ファイルサイズ＞（サイズ指定）	logrotateコマンド実行時に、ログファイルが指定サイズを超えている場合。 • 切り替わる例 指定サイズが100KBで、コマンド実行時のログファイルサイズが100KB以上の場合。
コマンドオプション	[-f]オプション	logrotateコマンド実行時に、設定ファイルに記述されたすべてのシステムログファイルを切り替えます。 同日内（0時0分～23時59分）に2回以上実行した場合、ログファイルは切り替えません。

■ システムログファイルの切替え周期に関する注意事項

ログファイルの切替え周期の設定では、次のことに注意してください。

システムログファイルの切替え周期に合わせた**logrotate**コマンドの実施

logrotateコマンドは、コマンドが実行された時に設定ファイルに記述されたログファイルの切替え周期を確認します。そのため、設定ファイルに記述した最も短い周期に合わせてlogrotateコマンドを実行する必要があります。例えば、daily指定とweekly指定のログファイルがあった場合、logrotateコマンドを毎日実行しなければdaily指定の意味がありません。このような状態にならないよう、logrotateコマンドを実行するタイミングを設計してください。

システムログファイルの強制切替え

[-f]オプションを指定したlogrotateコマンドを実行した場合、設定ファイルに記述されているすべてのシステムログファイルが切り替わります。

設定ファイル内のすべてのログファイルを切り替えることが運用上適切でない場合、強制切替え対象とするログファイルだけの設定を記述した設定ファイルを新たに作成します。ログファイルの強制切替えが必要な場合、ここで作成した設定ファイルを使用して、logrotateコマンドを実行します。

切替え周期内に複数回の**logrotate**コマンドを実行

ログファイルの切替え周期をmonthly/weekly/dailyに設定している場合、ログファイルを切り替えたあと、切替え周期内にlogrotateコマンドを実行しても、ログファイルの切替えは発生しません。daily指定ならば日が、weeklyならば週が、monthlyならば月が変わったあとのlogrotateコマンド実行時にログファイルが切り替わります。

ファイルサイズ指定の切替え

ファイルサイズ指定によるログファイルの切替えの場合、logrotateコマンドの実行時にログファイルが指定サイズ以上であればログファイルが切り替わります。logrotateコマンドの実行時に指定サイズに達していない場合、その後の事象ログデータの書込みでログファイルが指定サイズ以上になってもログファイルの切替えは発生しません。そのため、logrotateコマンドの実行タイミングによっては、指定サイズを大幅に超えるログファイルが作成される場合がありますので、注意してください。

4.2.3 システムログファイルのバックアップに関する注意事項

過去のシステム稼働状況や障害発生状況を調査する場合に備えて、システムログファイルを定期的にバックアップすることを推奨します。

システムログファイルのバックアップは、システム運用の一環として計画し、システムログファイルの切替えに関連させて実施します。

バックアップするログファイルは、ログファイルの切替えが終わり、事象ログデータが書き込まれていない状態のログファイル（以降、世代ログファイルと表記します）とします。

使用中（事象ログデータが書き込まれている状態）のログファイルをバックアップすると、バックアップしている間に事象ログデータが書き込まれるため、バックアップデータの内容が保証できません。

- 切替えが終わったログファイルは、世代ファイルとして次のように管理されます。この管理方法を前提として、世代ファイルのバックアップ運用を設計します。

  また、「4.3.2 logrotateコマンドのログファイル切替え方式」（→P.73）も併せてお読みください。

  - 世代ファイルは、使用中のログファイルと同じディレクトリ配下に作成されます。
  - 世代ファイルのファイル名は、システムログファイル名 + "-日付"となります。
  - 保存される世代ファイル数は、設定ファイルに記述された保有世代数が上限となります。
  - 世代ファイルが、設定ファイルに記述された保有世代数を超える場合、最も日付の古い世代ファイルが削除されます。

4.2.4 ログローテーションのスケジュール変更

ここでは、logrotateスクリプト実行スケジュールの変更方法について説明します。

logrotateスクリプトをrootユーザー用のスケジュールとして設定する例を示します。

1. /etc/cron.dailyディレクトリ配下にあるlogrotateスクリプトを、root権限でアクセスできるディレクトリ配下へ移動します。

   移動先ディレクトリは、/tmpディレクトリ配下などの定期的にファイルが削除されるディレクトリは避けてください。

   例：/root/dailyディレクトリで運用する場合

   ```
   # mv /etc/cron.daily/logrotate /root/daily
   ```

2. logrotateスクリプトの実行スケジュールを編集します。

   ```
   # crontab -e
   ```

このとき使用されるエディタは、環境変数EDITORまたはVISUALに設定されているエディタが使用されます。環境変数が設定されていない場合、viが使用されます。

3. logrotateスクリプトの実行スケジュールを設定します。

例：毎日0時30分に実行する場合

```
     :
30 0 * * * /root/daily/logrotate
     :
```

4. 変更後の設定内容を確認します。

```
# crontab -l
```

4.2.5 ログローテーション条件の変更

ログファイルの切替え周期および保有世代数の変更は、ログファイルに対応した設定ファイルを編集することで実現します。

設定ファイルの記述については、「4.4.2 ログローテーション運用の設定ファイルの記述様式」（→P.79）をお読みください。

1. root権限で、ログファイルに対応する設定ファイルを編集します。

例：namedデーモンのログ設定ファイルをviで編集する場合

```
# vi /etc/logrotate.d/named
```

例：ログファイルの切替えを毎日行う設定の場合

```
/var/log/named.log {
    daily
    missingok
    create 0644 named named
    postrotate
        /sbin/service named reload 2> /dev/null > /dev/null
    endscript
}
```

2. 設定ファイルの記述誤りがないことを確認するため、logrotateコマンドをデバッグモードで実行します。

```
# logrotate -d /etc/logrotate.conf 2>&1
```

記述誤りがある場合、「error:～」の警告メッセージが表示されますので、該当箇所を修正します。

4.3 ユーザー固有ログファイルのログローテーション

ユーザー固有ログファイルのログローテーションについて説明します。

4.3.1 ログローテーションの検討事項

ユーザー固有ログファイルのログローテーションを設計するに際して、決めなければならない項目について説明します。

対象ログファイル

対象とするログファイル名を決定します。

実行ユーザー

ユーザー固有ログファイルのパーミッションに基づき、logrotateスクリプトを実行するユーザーを決定します。

- 実行ユーザーがroot権限の場合

  システムログファイルのログローテーションに追加することができます。この場合のログファイル切替え設定については、「4.3.3 システムログのログローテーションへの追加」（→P.76）をお読みください。

- 実行ユーザーがroot権限ではない場合、またはroot権限で実行するが独自のログローテーションを行う場合

  利用するコマンドおよびスケジューリングシステムの仕様に基づき、ログローテーションを設計します。

  logrotateコマンドを独自に使用する場合、または別の手段を使用してcronでスケジューリングを実施する場合の設定方法については、「4.3.4 logrotateコマンドを独自に使用する」（→P.77）をお読みください。

ログローテーション実現コマンド

logrotateコマンドは、システム標準のログローテーションを実現するコマンドです。しかし、何らかの事情でlogrotateコマンドを使うことができない場合、ログローテーションを実現するコマンドを選定し、ログファイル切替えの方式を検討します。

logrotateコマンドを使用する場合は、logrotateコマンドのログファイル切替え方式に合わせて、アプリケーションを設計します。

logrotateコマンドのログファイル切替え方式については、「4.3.2 logrotateコマンドのログファイル切替え方式」（→P.73）をお読みください。

4.3.2 logrotateコマンドのログファイル切替え方式

ここでは、logrotateコマンドが提供するログファイル切替え方式を説明します。

logrotateコマンドを使用してユーザー固有ログファイルのログローテーションを実現する場合、アプリケーションの構造に適した方式を選択します。

■ ログファイル切替え方式

logrotateコマンドの提供するログファイル切替え方式には、次の方式があります。

- ログファイルの実体を切り替えて保存する方式
- ログファイルのコピーを保存する方式

■ ログファイルの実体を切り替えて保存する方式

この方式では、アプリケーションがlogrotateコマンドから任意のシグナル通知を受ける仕組みが必要です。シグナル通知を受けたアプリケーションは、ログファイルをオープンし直すことで、ログデータを書き込むファイル実体を変更します。

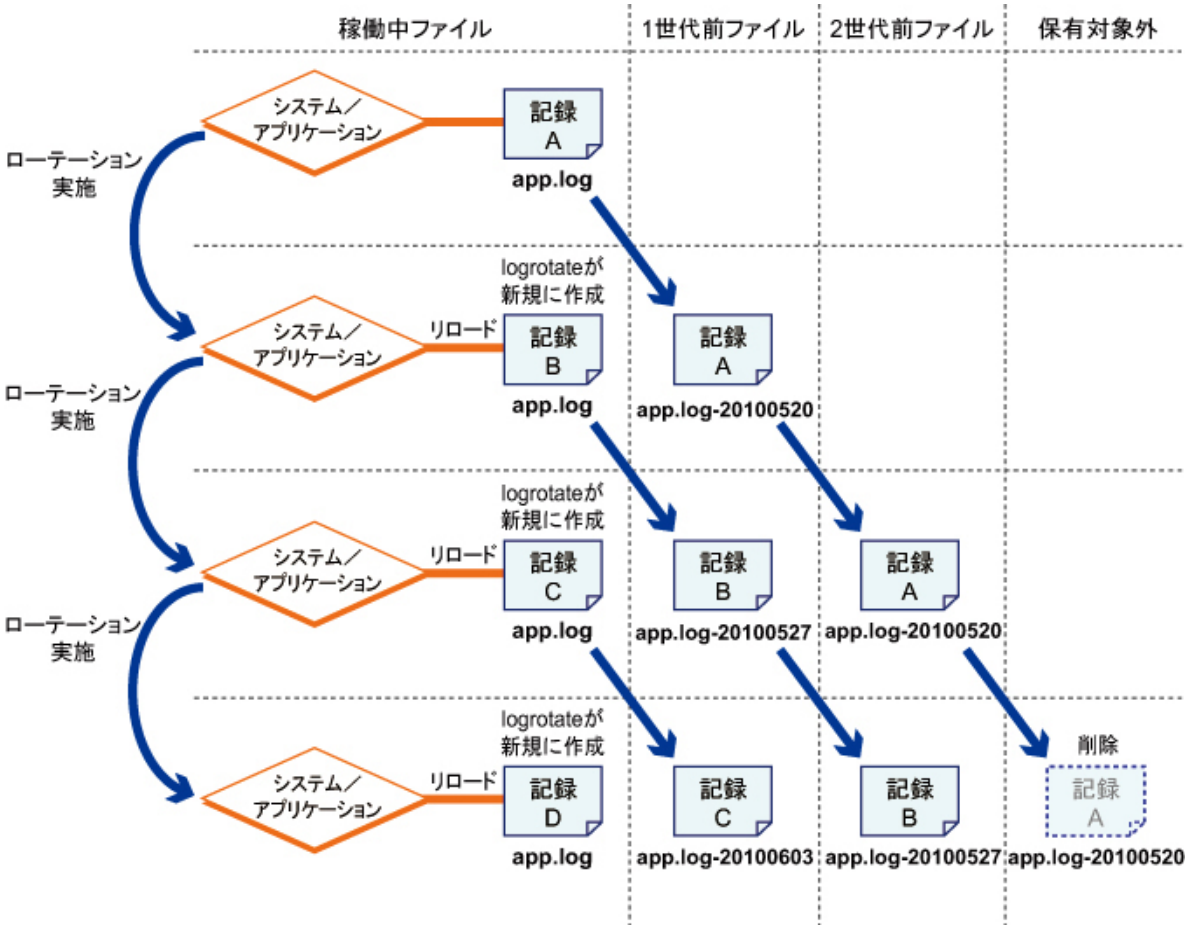
アプリケーションがシグナル通知を受ける仕組みがない場合でも、ログファイル切替えの場合にアプリケーションの再起動が許されるならば、この方式を使用できます。

次に、logrotateコマンドによるログファイル切替えの動作を説明します。

1. 現在使用中のログファイル名を変更し、“-”と日付を付与したファイル名にする。
2. ログデータを書き込むファイル名の新規ファイルを作成する。
3. アプリケーションに任意のシグナルを送信し、ログファイルの再オープンを通知する。
4. 保有世代数を超えたログファイル（最も日付の古いログファイル）を削除する。

次に、ログローテーションの動作を示します。例は、ログファイル名app.logの切替え周期がweeklyかつ保有世代数が2世代のログファイルapp.logを保持するよう設定した場合とします。

図 21 : ファイル実体を替える方式によるログローテーションの動作



このように、保有世代数を超える日付の古いログファイルは、ログファイル切替えのたびに削除されます。すべてのログデータを保存するためには、ログファイルが削除される前、つまり次回の ログローテーション開始までの間に、古いログファイルのバックアップを完了させておきます。バックアップの時期は、ログローテーションによりログファイルを切り替えた場合を推奨します。

- logrotateコマンド設定ファイルには、createディレクティブを記述します。

■ ログファイルのコピーを保存する方式

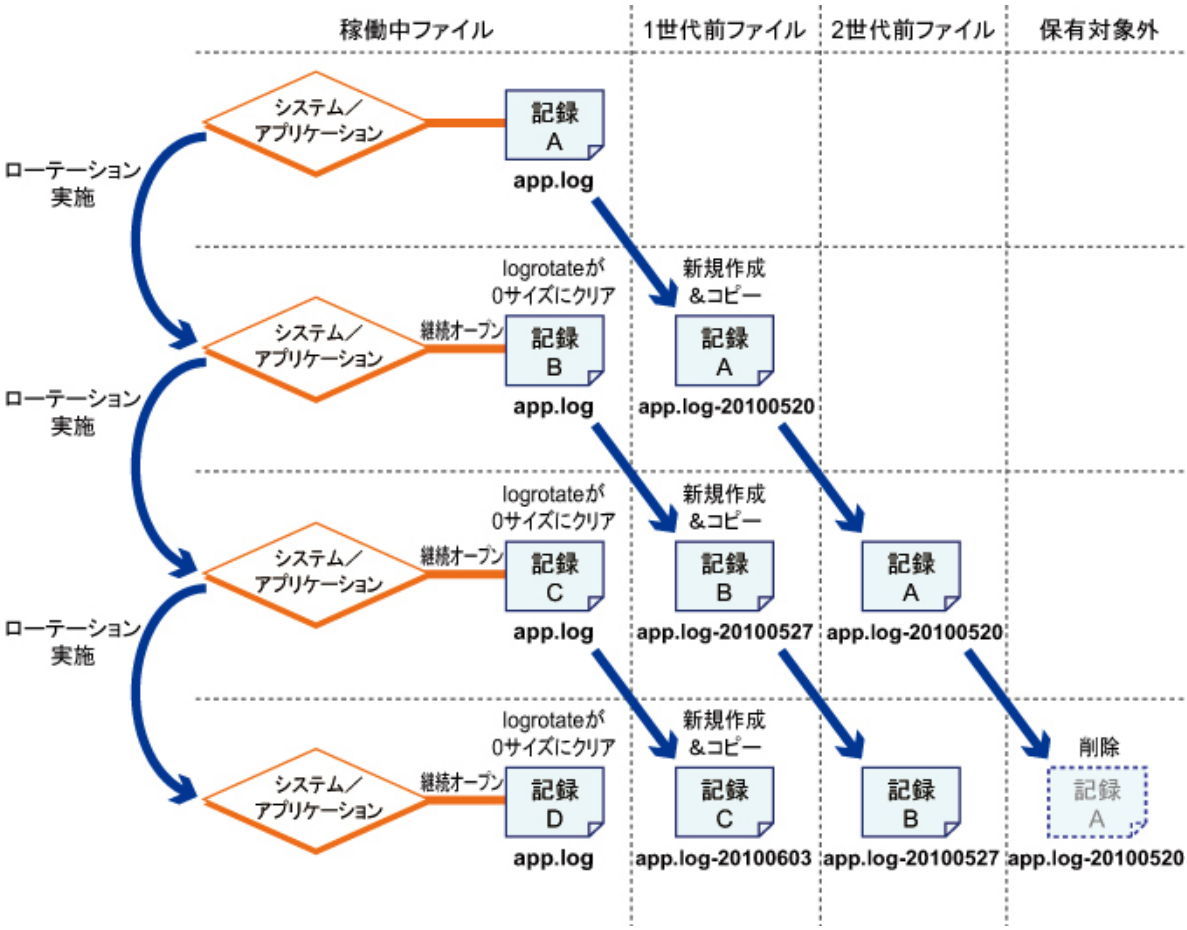
ログファイルの実体を切り替えて保存する方式が使えない場合に使用しますが、ログデータの一部を失う可能性があります。そのため、この方式を使用することは推奨しません。

次に、logrotateコマンドによるログファイル切替えの動作を説明します。

1. ログファイル名に“-”と日付を付与したファイル名の新規ファイルを作成する。
2. 現在使用中のログファイルをオープンし、すべてのデータを処理1.で作成したファイルにコピーする。
3. 現在使用中のログファイルサイズを0サイズに設定（truncate）する。
4. 保有世代数を超えたログファイル（日付の一番古いログファイル）を削除する。

次に、ログローテーションの動作を示します。例は、ログファイル名app.logの切替え周期がweeklyかつ保有世代数が2世代分のログファイルapp.logを保持するよう設定した場合とします。

図 **22 :** ログファイルのコピーを保存する方式によるログローテーションの動作



このように、保有世代数を超える日付の古いログファイルは、ログファイル切替えのたびに削除されます。すべてのログデータを保存するためには、ログファイルが削除される前、つまり次回のログローテーション開始までの間に、古いログファイルのバックアップを完了させておきます。バックアップの時期は、ログローテーションによりログファイルを切り替えた場合を推奨します。

注 意

- この方式は、ログファイルのコピー終了後からログファイルサイズを0サイズにするまでの間に、アプリケーションからログデータが書き込まれると、そのログデータが失われます。

参 考

- 設定ファイルには、copytruncateディレクティブを記述します。

4.3.3 システムログのログローテーションへの追加

root権限でログローテーションを実行する場合、システムログのログローテーションにユーザー固有ログファイルを追加できます。

1. ユーザー固有ログファイル用の個別設定ファイルを作成します。
   個別設定ファイルのファイル名は、対応するログファイルが連想できるファイル名にします。

ここでは、/etc/logrotate.dディレクトリ配下にuser.confファイルを作成する場合を例に説明します。

個別設定ファイルには、ログファイルの絶対パス名を記述します。

記述例：ユーザー固有ログファイルの絶対パス名が/home/user/user.logの場合

```
/home/user/user.log {
    nocompress
    missingok
}
```

- 個別設定ファイルに記述する切替え周期、保有世代数などを省略した場合、標準設定ファイル/etc/logrotate.confに記述されている設定がログファイル共通の設定として継承されます。
- 設定ファイルの記述方法については、「4.4.2 ログローテーション運用の設定ファイルの記述様式」（→P.79）をお読みください。

2. 作成した個別設定ファイルを/etc/logrotate.dディレクトリ配下に配置します。

配置後、ファイルの読込み権限を644に設定します。

```
# chmod 644 /etc/logrotate.d/user.conf
```

3. 設定ファイルの記述誤りがないことを確認するため、logrotateコマンドをデバッグモードで実行します。

```
# logrotate -d /etc/logrotate.conf
```

- 「reading config file ＜設定ファイル名＞」という表示が続きます。配置したファイルが表示されることを確認します。
- 記述誤りがある場合、「error:～」の警告メッセージが表示されますので、該当箇所を修正します。

4.3.4 logrotateコマンドを独自に使用する

logrotateコマンドを独自に使用する場合の設定手順について説明します。

1. ユーザー固有ログファイルに対応する個別設定ファイルを作成します。

個別設定ファイルは、任意の場所に作成できますが、/tmpディレクトリ配下などの定期的にファイルが削除されるディレクトリ配下は避けてください。設定ファイルの記述方法については、「4.4.2 ログローテーション運用の設定ファイルの記述様式」（→P.79）をお読みください。

注 意

- 個別設定ファイルは、標準設定ファイル（/etc/logroatate.conf）内の#include文で指定しているディレクトリ以外を指定してください。標準設定ファイルについては、「4.4.1 ログローテーション設定ファイルの種類」（→P.78）をお読みください。

2. 状態ファイルを作成するディレクトリおよび状態ファイルのファイル名を決定します。

ディレクトリおよびファイルは、logrotateコマンドが作成しますので、あらかじめ作成する必要はありません。

参 考

- 状態ファイルは、ログファイルを切り替えた最終日付を記録するファイルです。logrotateコマンドの[-s]オプションに絶対パス名で指定します。[-s]オプションを指定しない場合、デフォルトの/var/lib/logrotate.statusファイルが使用されます。

3. 設定ファイルの記述誤りがないことを確認するため、logrotateコマンドをデバッグモードで実行します。

```
# logrotate -d -s ＜状態ファイル絶対パス名＞ ＜設定ファイル絶対パス名＞
```

- デバッグモードで実行した場合、状態ファイルは作成されません。
- 記述に誤りがある場合、「error:～」の警告メッセージが表示されますので、該当箇所を修正します。

4. logrotateコマンドを実行し、ログファイルの切替えが行われることを確認します。

```
# logrotate -s ＜状態ファイル絶対パス名＞ ＜設定ファイル絶対パス名＞
```

5. スケジュール運用を行う場合は、手順4で実行したコマンドを登録します。

cronでスケジュール運用を行う場合は、「4.2.4 ログローテーションのスケジュール変更」（→P.71）の作業手順に沿って登録します。

注意

- 個別にユーザー固有ログファイルをログローテーションする場合、状態ファイルは、システムのデフォルトを使わないで、別のファイルを使用してください。

4.4 ログローテーション設定ファイル

logrotateコマンドを使用する際の設定ファイルの種類と記述様式について説明します。

4.4.1 ログローテーション設定ファイルの種類

logrotateコマンドは、コマンド実行時に設定ファイルを読み込むことで、ログローテーションの実施条件を取得します。

logrotateコマンドが使用する設定ファイルは、次の2種類があります。

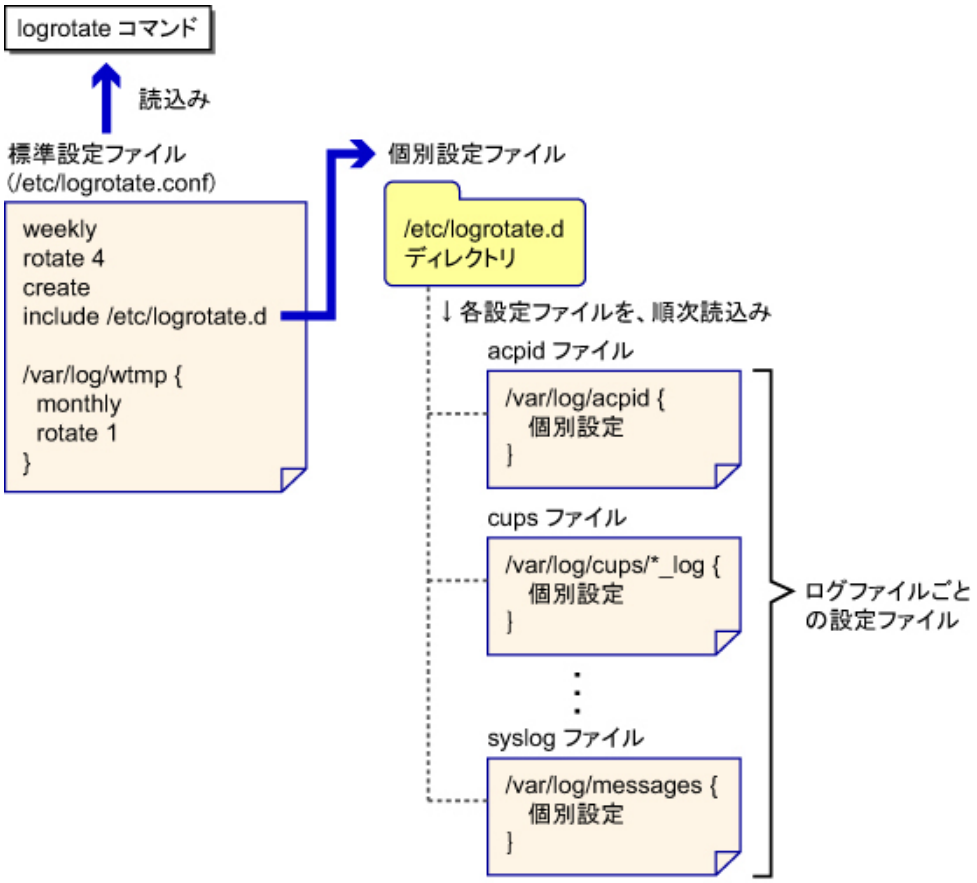
- 標準設定ファイル（/etc/logrotate.conf）

  logrotateコマンド実行時に必ず読み込む設定ファイルです。この設定ファイルの内容が、すべてのログローテーションのデフォルト設定となります。

- 個別設定ファイル（/etc/logrotete.dディレクトリ配下）

  個々のログファイルに対するログローテーションの実施条件を設定するファイルです。

図 **23 : logrotate**コマンドと各設定ファイルの関係



4.4.2 ログローテーション運用の設定ファイルの記述様式

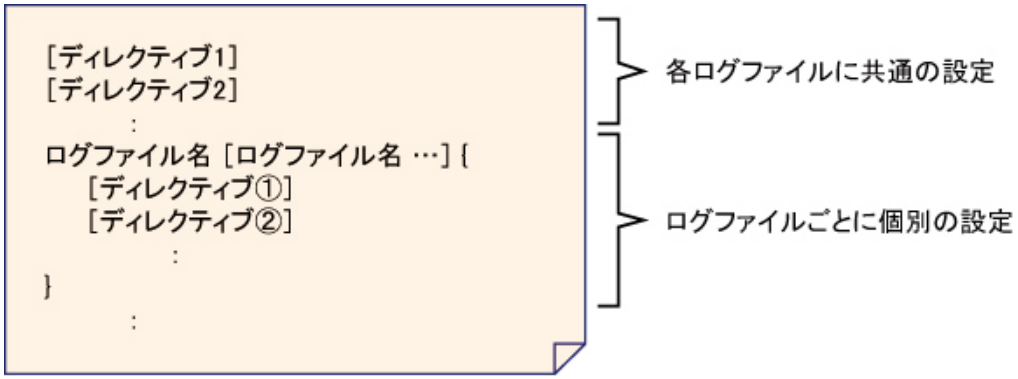
設定ファイルには、ログファイルの切替え周期、世代管理の制御を行うディレクティブなどを記述します。

参考

- ディレクティブについては、オンラインマニュアルでlogrotateコマンドの仕様を参照してください。

■ logrotateコマンドの設定ファイル書式

図 **24 : logrotate**コマンドの設定ファイル書式

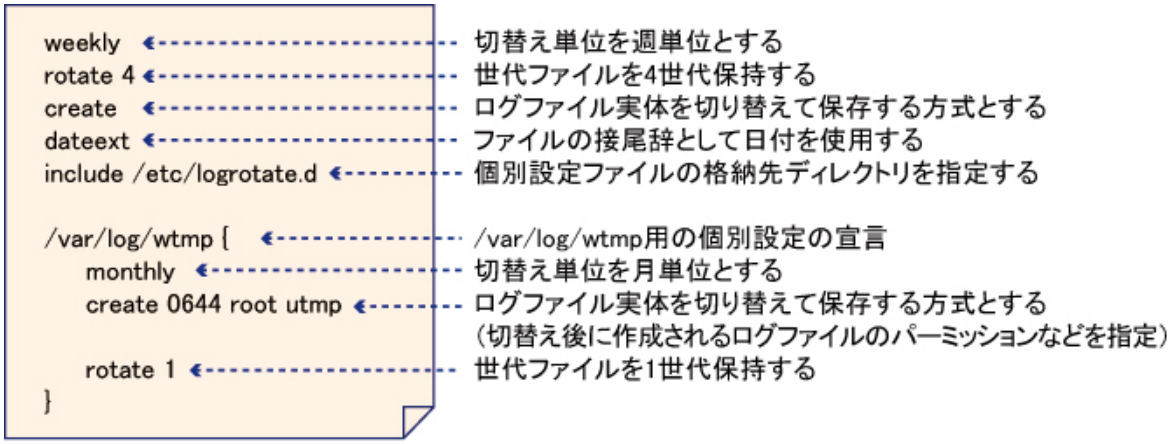


設定ファイルには、次の決まりがあります。

- すべてのログファイルに共通の設定を先に記述し、そのあとにログファイルごとの個別設定を記述します。
- 共通設定で設定した項目は、個別設定に記述することによって上書きされます。
- ログファイル名には、複数のファイル名を併記する、またはメタキャラクタ（*）を使用できます。この場合、該当する複数ファイルに対して共通の設定が適用されます。
- 「#」で開始する行はコメント行になります。

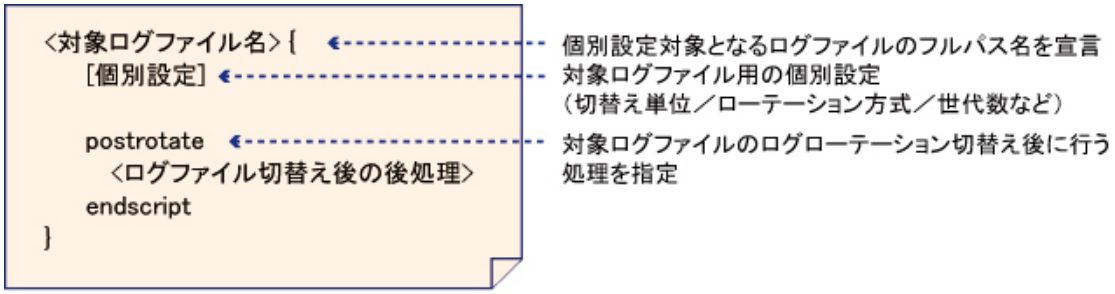
■ 標準設定ファイル（/etc/logrotate.conf）の記述例

図 **25 :** 標準設定ファイル（**/etc/logrotate.conf**）の記述例



■ 個別設定ファイルの書式

図 **26 :** 個別設定ファイルの書式



「postrotate」は、ログファイルの実体を切り替えて保存する方式を使ってログファイルを切り替える場合に指定します。

「＜ログファイル切替え後の後処理＞」は、ログファイルをオープンし直すことでファイル実体を変更する指示を出す操作（シグナルの送信など）を記述します。

- アプリケーションを停止しない場合の後処理

  アプリケーション内でHUP シグナルをハンドリングし、ログファイルのクローズおよび再オープンを行い、ログデータを書き込むファイルの実体を切り替えます。

```
postrotate
     kill -HUP `cat /home/user/user.pid`
endscript
```

  /home/user/user.pidファイルには、アプリケーションのプロセスIDが設定されているものとします。

- アプリケーションを停止する場合の後処理

  アプリケーションを再起動することで、ログファイルのクローズおよび再オープンを行い、ログデータを書き込むファイルの実体を切り替えます。

```
postrotate
     /home/user/userapply restart`
endscript
```

  /home/user/userapply はアプリケーションの絶対パス名です。アプリケーションは[restart]オプションにより再起動します。

注 意

- ログファイルのログローテーション定義が複数の個別設定ファイルに存在する場合、ログファイルの切替え処理が失敗します。実際に使用しない個別設定ファイル（過去の設定の保存などを含む）は、削除または個別設定ファイル配置ディレクトリから移動してください。
- logrotateコマンド実行時に個別設定ファイルに定義されているログファイルが存在しない場合、ログファイルの切替え処理が失敗します。ログファイルが存在しない場合は、事前にログファイルを作成またはログデータの書込みによりログファイルが作成された後にlogrotateコマンドを実行してください。

5 システム設計：時刻同期（NTP）

ここでは、RHELシステムにおける時刻の仕組み、時刻同期の必要性、および時刻同期に重要な役割を果たすNTP（Network Time Protocol）サービスについて説明します。

5.1 時刻の仕組みと時刻同期

RHELシステムが使用する時計には、ハードウェア時計とソフトウェアが維持するシステム時計があります。それぞれの時計に対する時刻同期について説明します。

5.1.1 時刻同期の必要性

時刻補正が必要な理由について説明します。

コンピュータは、その性能にかかわらず、内蔵されている水晶発振器によって時刻を維持しています。この水晶発振器は個体ごとに誤差があるため、時間が経過することによる誤差の累積により、正確な時刻を維持できなくなります。この状態を放置すると、次のような影響が生じます。

- イベントログの時刻ずれ
  イベントログの記録においてタイムスタンプが付与される場合、正確な時刻が記録されません。
- システム間連携における障害発生
  時刻が一致しないシステム間で、タイムスタンプを使用するデータを連携する場合、時刻のずれによりシステムがエラーと判定する事象が発生する場合があります（ネットワーク共有しているファイルの更新時刻が、過去の時刻で更新されるように見えるなど）。

これらのことから、システムの時刻を正確に保つことが必要となってきます。

5.1.2 ハードウェア時計とシステム時計

ハードウェア時計とシステム時計の関係について説明します。

ハードウェア時計は、ハードウェアに搭載されている時計です。リアルタイムクロックとも呼ばれ、サーバの電源切断後も、バッテリーにより駆動され、時刻を維持します。

システム時計は、カーネルがタイマー割込みを使用して時刻を維持するソフトウェア時計です。RHELシステムで使用される時刻は、システム時計が使用されます。

ハードウェア時計とシステム時計は、次のように関係します。

- RHELシステム起動時
  カーネルがハードウェア時計から時刻を取得し、システム時計に設定します。
- RHELシステム停止時
  カーネルが維持してきたシステム時計の時刻を使用して、ハードウェア時計を設定します。

■ ハードウェア時計の設定

RHELシステムを長期間稼働する（システムの停止および起動が発生しない）場合、ハードウェア時計とシステム時計の同期をとる契機がありません。そのため、ハードウェア時計とシステム時計の時刻は、わ

ずかなずれが累積し、秒単位から分単位の時刻のずれにつながっていきます。この時刻のずれは、ハードウェアとRHELの各イベントログのタイムスタンプに影響します。例えば、トラブル発生時の原因究明において、RHELとハードウェアのそれぞれで発生したイベントの時系列上の比較が困難になります。そのため、システムを長期間稼働する場合は、システム時計の時刻を使用して、定期的にハードウェア時計を設定することを推奨します。

ハードウェア時計の設定

root権限で、システム時計の時刻をハードウェア時計に設定します。

```
# hwclock --systohc
```

システム時計とハードウェア時計の時刻差の確認

root権限で、ハードウェア時計とシステム時計の時刻差を確認します。

```
# hwclock --show
＜曜日＞＜月＞＜日＞＜時刻＞＜年＞＜システム時計との時刻差＞ seconds
```

コマンド実行例を示します。

```
Tue Oct 19 08:44:24 2010  -0.591105 seconds
```

表示される時刻は、ハードウェア時計の時刻です。そのあとに表示されている秒数は、システム時計との差を示しています。マイナス値は、ハードウェア時計がシステム時計よりも進んでいることを意味します。時刻差が大きい場合、「ハードウェア時計の設定」を実行します。

■ システム時計の補正

正確な時刻を保つためには、システム時計の補正が必要です。システム時計の補正は、NTP運用を推奨します。NTP運用については、「5.2 NTPの時刻同期方式」（→P.83）以降で詳しく説明します。

5.2 NTPの時刻同期方式

ここでは、NTPによるシステム時計の補正について説明します。

5.2.1 NTPとは

NTP（Network Time Protocol）とは、ネットワークに接続されるシステムにおいて、OSが維持するシステム時計を正確な時刻と同期するためのプロトコル（規格）です。

NTP運用では、ネットワーク上に基準となる正確な時刻情報を維持するサーバを設置し、NTPサーバとします。時刻をNTPサーバの持つ時刻と同期するサーバは、NTPクライアントを使うことで、NTPサーバから正確な時刻情報を入手し、そのシステムのシステム時計を補正します。

- NTPは、OSI基本参照モデルの第7層（アプリケーション層）に位置し、ネットワーク通信の待ち受けにUDPポートの123番を使用します。ファイアーウォール設定でUDPポートの123番の通信が遮断されている場合は、遮断設定を解除します。

■ NTPの利点

サーバの時刻を正確な時刻と同期するために、dateコマンドを手動で実行する場合、次の問題が発生します。

- 継続的な時刻の維持が困難
- 手作業に伴う誤差の発生

これらの問題は、サーバの時刻を自動的に同期する仕組みがあれば、解決できます。NTPはこれらの問題を排除するために設計され、サーバの時刻を正確な時刻と自動的に同期させます。

5.2.2 NTPの時刻同期の仕組み

NTPの時刻同期の仕組みについて、NTPサーバの階層構造と処理の流れを説明します。

■ 階層構造（stratum）

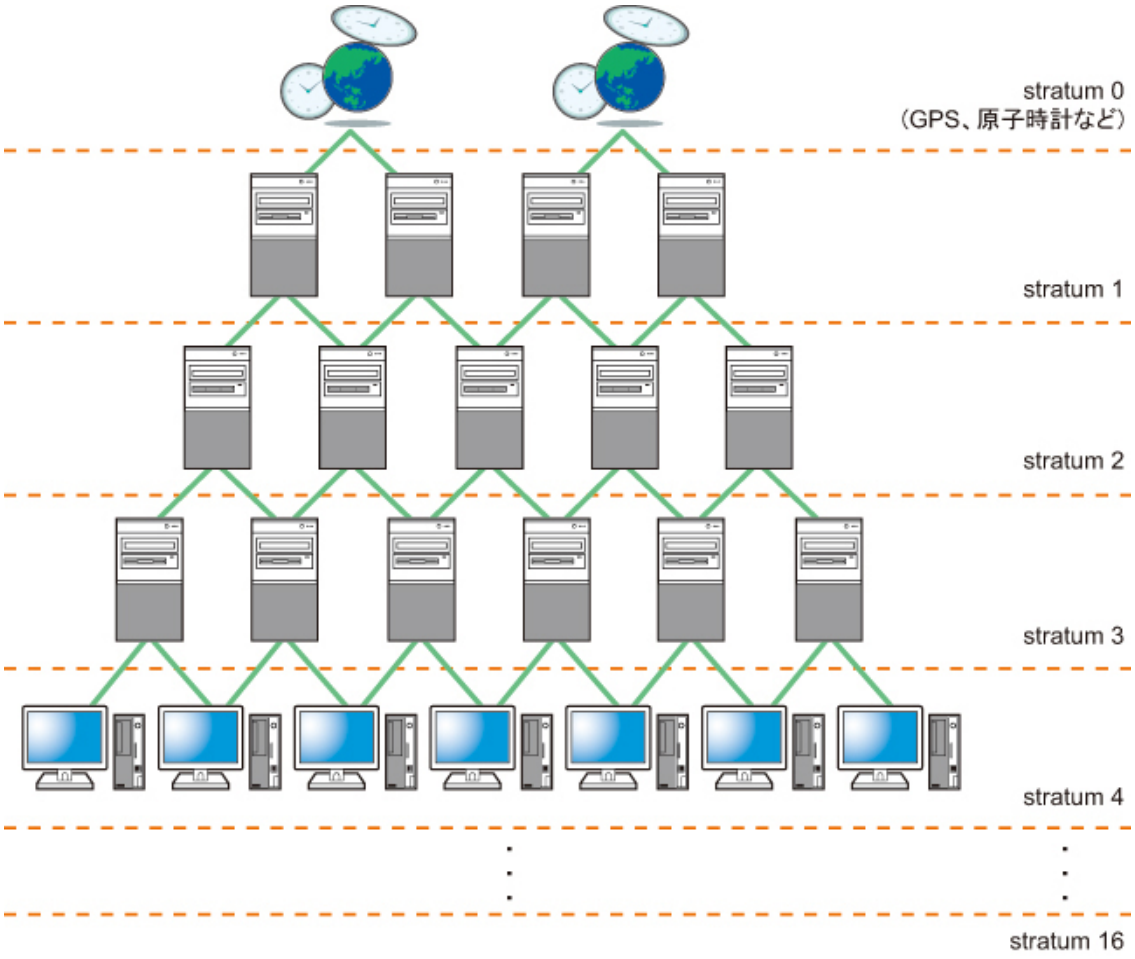
NTPのネットワークは、stratumと呼ばれる階層構造になっており、インターネット上でもイントラネット上でもstratum（階層）の番号が小さいほど正確な時計（UTC）に近いことを表しています。最上位のNTPサーバが正確な時計を参照することで時刻情報を取得します。下位のNTPサーバは、上位のNTPサーバから時刻情報を取得することで時刻を同期します。最上位のNTPサーバはstratum 1となり、階層が下がるごとに数字が1つずつ大きくなります。最下位のNTPサーバはstratum 16です。同じ階層のNTPサーバ間では、時刻情報の取得はできません。

- 原始時計またはGPSは、stratum 0の位置づけとなります。

次図に、時刻同期を行うNTPサーバのstratum階層の概念図を示します。階層構造は、サーバ機の業務上の関係性を考慮して構成します。例えば、ファイル共用サービスにおいて、ワークステーションの時刻をファイルサーバの時刻と同期させる場合、ファイルサーバの階層がstratum 3ならば、ワークステーションの階層はstratum 4となります。

最下位はstratum 16なので、15階層までNTPサーバを設定できます。この階層関係を調整することで、NTP運用に伴う負荷分散を設計します。時刻精度の観点からは、stratumの大きさよりも、サーバとのネットワーク上での物理的な近さ（通過するルータの数など）の方が大きく影響します。

図 **27 : NTP**の階層構造



■ 時刻同期の流れ

次図に、NTPの時刻同期の流れを示します。

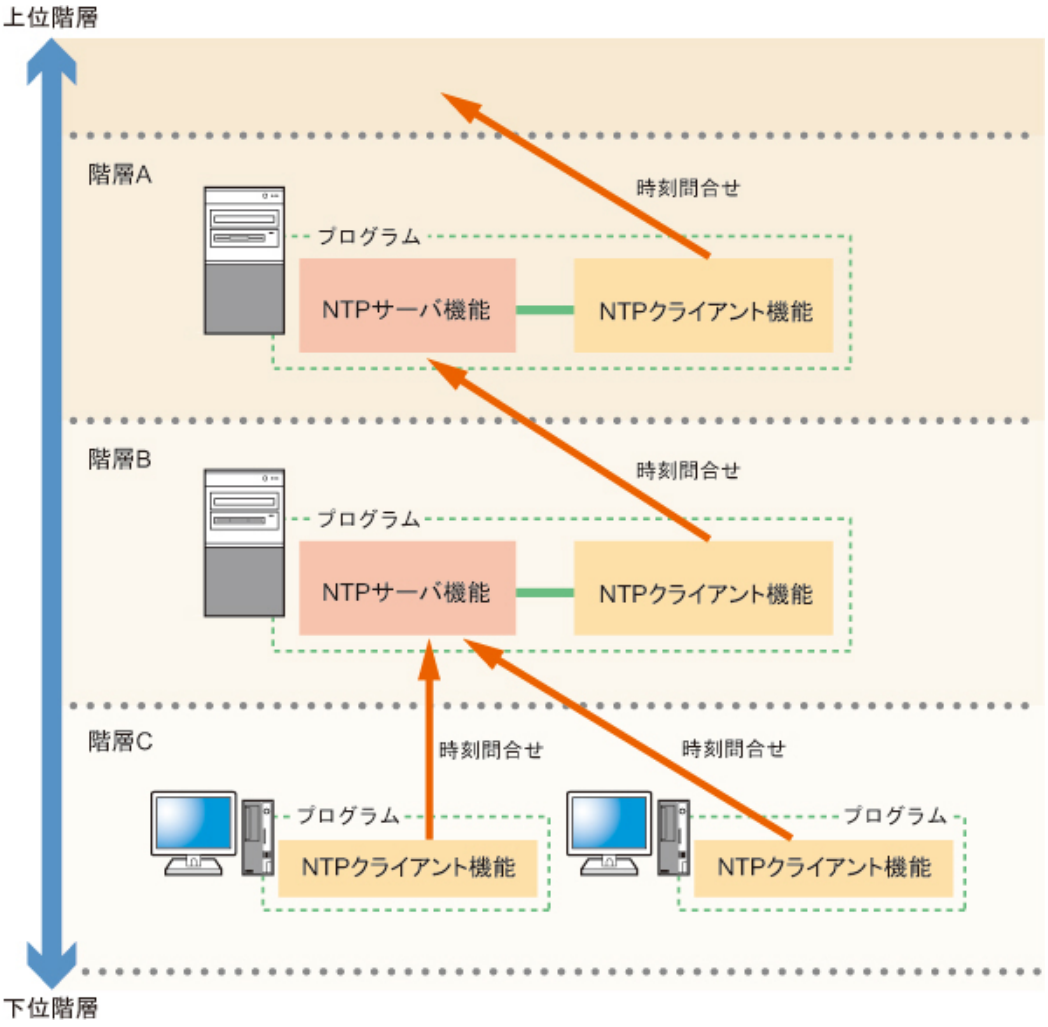
時刻同期をするサーバのNTPサーバ機能は、NTPクライアント機能を使用して上位のNTPサーバに接続して、自サーバの時刻を同期させます。

時刻情報を取得したNTPサーバ機能は、時刻を上位NTPサーバと同期させるだけでなく、自サーバより下位のNTPサーバに時刻情報を提供するNTPサーバになることができます。この場合、stratumの値は上位サーバのstratum値よりも1大きくなります。

参考

- NTPクライアントは、ネットワークトラフィックの増減による上位のNTPサーバからの応答時間のばらつきを補正し、取得した時刻情報にミリ秒単位の精度を与えます。そのために、NTPクライアントは、上位NTPサーバとのネットワーク通信情報を計測し続けています。

図 **28 :** 時刻同期の流れ



■ ローカルクロック

外部ネットワークから遮断されたネットワーク環境において、ネットワーク内のサーバの時刻を同期させる場合、時刻の基準となるシステムを用意します。このとき、時刻を問い合わせる上位NTPサーバが使えないため、時刻の問い合わせ先を自システムとし、自システムのシステム時計と同期させる形態をとります。NTP運用では、自システムのシステム時計をローカルクロックと呼びます。

また、NTP運用では、外部のNTPサーバとローカルクロックを併用することができます。この場合、外部NTPサーバの持つ時刻と同期することが優先されます。しかし、外部NTPサーバが停止している場合、またはネットワークの問題などで外部NTPサーバから応答がない場合は、ローカルクロックとの同期に切り替わります。このとき、外部NTPサーバのstratum値はローカルクロックに設定したstratum値より1大きくなります。ローカルクロックの指定方法およびstratum値の設定については、「5.4.1 NTP設定ファイル」（→P.97）をお読みください。

■ peer設定

NTP運用では、上位のstratumを持つNTPサーバとの時刻同期設定に加え、同じstratum階層のNTPサーバ間での時刻同期設定ができます。この設定方法をpeer設定と呼びます。

peer設定での必須条件

peer設定を行う場合、次の条件をすべて満たさなければなりません。

- 同一stratum階層のNTPサーバであること。
- peer設定によってペアを組む同一階層のNTPサーバは、それぞれが別の上位NTPサーバを参照すること。

注 意

- peer設定における必須条件を満たさない場合、peer設定されたNTPサーバ間で時刻ずれが発生します。

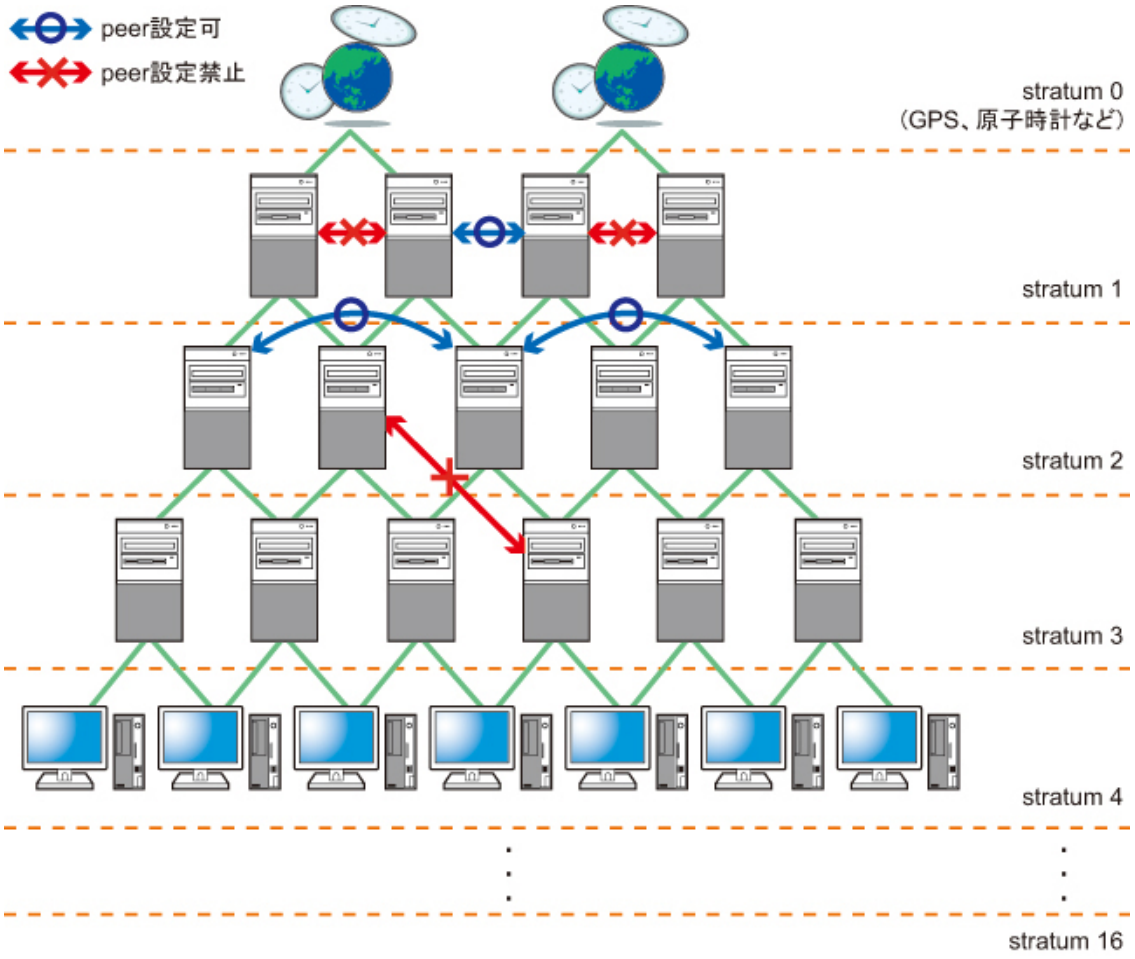
peer設定の効果

peer設定により、次の効果を得ることができます。

- peer設定をしたNTPサーバ間で、時刻の差が収束されます。これにより、同一stratum階層内のNTPサーバの時刻精度が均一化されます。
- peer設定をしたNTPサーバのどれかが上位NTPサーバと通信できなくなった場合も、peer設定されたNTPサーバと同期することにより、時刻精度が維持されます。

次図に、peer設定が可能なNTPサーバと設定が禁止されるNTPサーバの概念を示します。

図 **29 : peer設定が可能なサーバと禁止されるサーバ**



peer設定の方法については、「5.4.1 NTP設定ファイル」（→P.97）をお読みください。

5.2.3 RHEL6における時刻同期機能

RHELには、NTPサーバおよびNTPクライアントとしてntpd、NTPクライアントとしてntpdateコマンドがあります。

■ ntpd

ntpdは、次の機能を持つデーモンです。

- NTPクライアントとして、上位stratumまたはpeer設定のNTPサーバ（以降、参照するNTPサーバと記述します）との時刻同期およびシステム時計を補正します。
- NTPサーバとして、下位またはpeer設定のstratumのNTPクライアントへ時刻情報を提供します。

ntpdは、複数の参照するNTPサーバと接続できます。ntpdは、参照するそれぞれのNTPサーバが持つ時刻の比較およびネットワークトラフィックの計測を行うことで、参照するNTPサーバの重み付けを行い、最も信頼できると判断した時刻を使用します。

■ ntpdサービス

ntpdサービスは、RHELシステム起動時にntpdを起動します。サービスの起動には、サービスの起動設定が必要です。

■ ntpdateコマンド

ntpdateコマンドは、参照するNTPサーバから時刻を取得して、システム時計を補正します。ntpdを使わない場合、または任意のタイミングでシステム時計を補正する場合に使用します。

■ ntpdateサービス

ntpdateサービスは、RHELシステム起動時にntpdateコマンドを実行します。サービスの起動には、サービスの起動設定が必要です。

- ntpdサービスおよびntpdateサービスが起動する設定の場合、ntpdateサービスが先に起動され、そのあとにntpdサービスが起動されます。

5.2.4 システム時計の補正

システム時計の補正方法について説明します。

ntpdは、参照するNTPサーバへの時刻問合せを64秒間隔で行います。この問合せ間隔をポーリング間隔と呼びます。ntpdは、時刻同期が完了すると、ネットワークのオーバーヘッドを減少させるために、ポーリング間隔を128秒、256秒、512秒、1024秒と伸ばしていきます。

システム時計の時刻補正には、次の2種類の調整方法があります。

Step調整

参照するNTPサーバから取得した時刻情報でシステム時計の時刻を、直接変更します。このとき、システム時計が参照するNTPサーバから取得した時刻より進んでいた場合、システム時計の時刻変更により、システム時計が過去の時間を示す（以降、時刻の逆進と記述します）可能性があります。

Slew調整

システム時計の時刻のずれをゆっくりと（1秒間に最大0.5ミリ秒ずつ）補正します。ntpdは、システム時計を進める場合に使う現時刻への加算値を調整することでシステム時計の時刻を補正します。加算値の調整は、参照するNTPサーバの時刻情報と比較することで、次のように調整します。

- NTPクライアント側が遅れている場合：増分値を大きくする（時間を進める）
- NTPクライアント側が進んでいる場合：増分値を小さくする（時間を遅らせる）
- 参照するNTPサーバと時刻が等しい場合：増分値は変更なし

■ システム時計の時刻補正モード

ntpdを使用してシステム時計の時刻補正を行う場合、Step調整とSlew調整を組み合わせて使用します。Step調整とSlew調整のタイミングは、ntpd起動時の[- x]オプションの指定により変更することができます。本書では、[-x]オプションを使用しないNTP運用を「Stepモード」、[-x]オプションを使用するNTP運用を「Slewモード」と記述します。

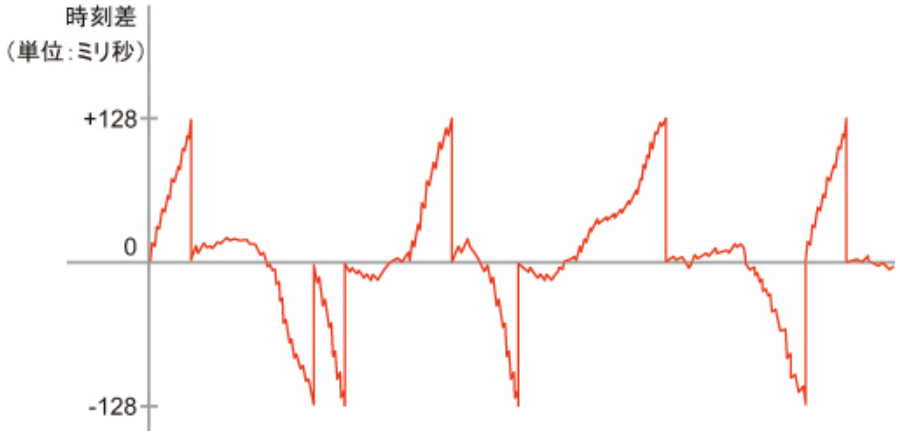
Stepモード

Stepモードは、ntpdのデフォルトの動作モードです。

参照するNTPサーバとの時刻差が128ミリ秒以内であればSlew調整を行い、128ミリ秒を超える場合にはStep調整を行います。具体的な調整は、次のように行います。

- 参照するNTPサーバとの時刻差が128ミリ秒以内の場合
  Slew調整により、システム時計の時刻を補正します。
- 参照するNTPサーバとの時刻差が128ミリ秒を超える場合
  参照するNTPサーバの時間情報を900秒間監視し、時刻差を確認します。
  900秒の監視の結果、±128ミリ秒を超える時刻のずれが解消されない場合、一時的に参照するNTPサーバとの時刻の同期を中断し、Step調整を行います。Step調整完了後、Slew調整を再開します。再開したStep調整では、最低5回の参照するNTPサーバへの時刻問合せによって、システム時計の時刻は参照するNTPサーバと同期状態となります（同期にかかる時間はポーリング間隔やその他の条件により異なります）。

図 30：Stepモードでの時刻補正



Slewモード

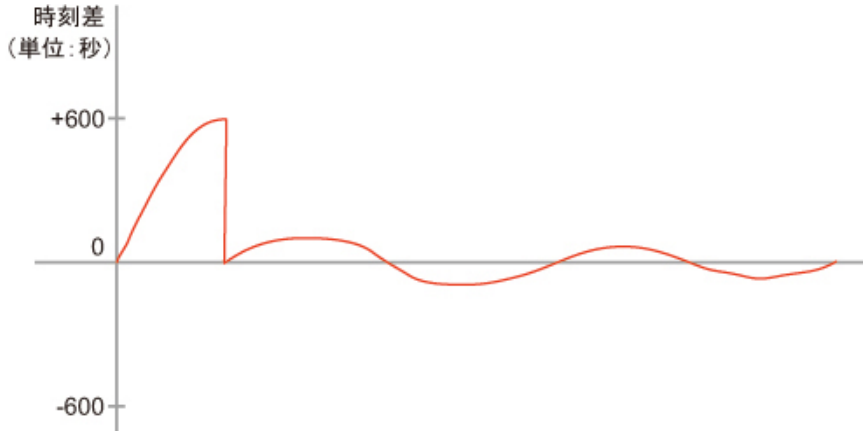
Slewモードも、Step調整とSlew調整を使い分けてシステム時計の時刻を補正します。Stepモードとの違いは、Slew調整を使う参照するNTPサーバとの時刻差にあります。

Slewモードは、参照するNTPサーバとの時刻差が600秒以内であればSlew調整を行い、600秒を超える場合にはStep調整を行います。

ntpdをSlewモードで運用する場合は、ntpdの起動オプションとして[-x]オプションを指定します。

ntpdの起動オプションの設定方法および確認方法については、「5.4.2 ntpdの起動オプション設定」（→P.103）をお読みください。

図 31：Slewモードでの時刻補正



注 意

- ntpdは、参照するNTPサーバの時刻情報との間に1000秒を超える時刻差を検出すると、次のメッセージをシステムログファイルに出力して終了します。

```
time correction of ＜秒数＞ seconds exceeds sanity limit (1000); set
 clock manually to the correct UTC time.
```

  ntpdateサービスの起動を設定した場合、ntpd起動前にntpdateコマンドによるシステム時計の時刻補正（Step調整）が行われます。そのため、RHELシステム起動時に、参照するNTPサーバとの時刻差が1000秒を超えていても、ntpdは直ちに終了しません。

Stepモード、Slewモードによる動作の違いは、次表のとおりです。

表 29：システム時計の時刻補正モードによるntpdの動作の違い

	Stepモード（デフォルト）	Slewモード
ntpd動作中	参照するNTPサーバとの時刻差により、次のように動作します。 • 128ミリ秒以内の場合 1秒間に最大0.5ミリ秒ずつシステム時計の時刻補正を行います。 • 128ミリ秒を超え、1000秒までの場合 システム時計の時刻を直接変更します。 • 1000秒を超える場合 システムログにメッセージを出力して終了します。	参照するNTPサーバとの時刻差により、次のように動作します。 • 600秒以内の場合 1秒間に最大0.5ミリ秒ずつシステム時計の時刻補正を行います。 • 600秒を超え、1000秒までの場合 システム時計の時刻を直接変更します。 • 1000秒を超える場合 システムログにメッセージを出力して終了します。

	Stepモード（デフォルト）	Slewモード
利点	参照するNTPサーバとの時刻同期が短時間で終了できます。	Stepモードに比べ、Slew調整する時間差の幅が広いため、時刻の逆進は発生しにくくなります。
欠点	時刻が逆進する場合があるため、データベース、時刻の逆進に対応していないアプリケーションなどに問題が発生する可能性があります。	1秒間に最大0.5ミリ秒までしか補正しないため、時刻の同期に時間がかかる可能性があります。

5.3 NTP環境の設計・導入

ntpdを使用したNTP環境の構築について、設計から導入までの流れを説明します。

5.3.1 NTP環境の設定手順

NTP環境の設定手順について説明します。

NTPにより時刻同期を行う場合には、次の流れでNTPの設定を行います。

「5.3.2 参照するNTPサーバの構成・参照設計」（→P.92）

⬇

「5.3.3 システム時計の時刻補正モード選定」（→P.93）

⬇

「5.3.4 RHELシステム起動時の初期参照NTPサーバの設計」（→P.94）

⬇

「5.3.5 ntpdの高度な動作設計」（→P.94）
ntpd 起動オプション、アクセス制御など必要に応じて実施します。

⬇

「5.3.6 NTP環境の導入」（→P.95）

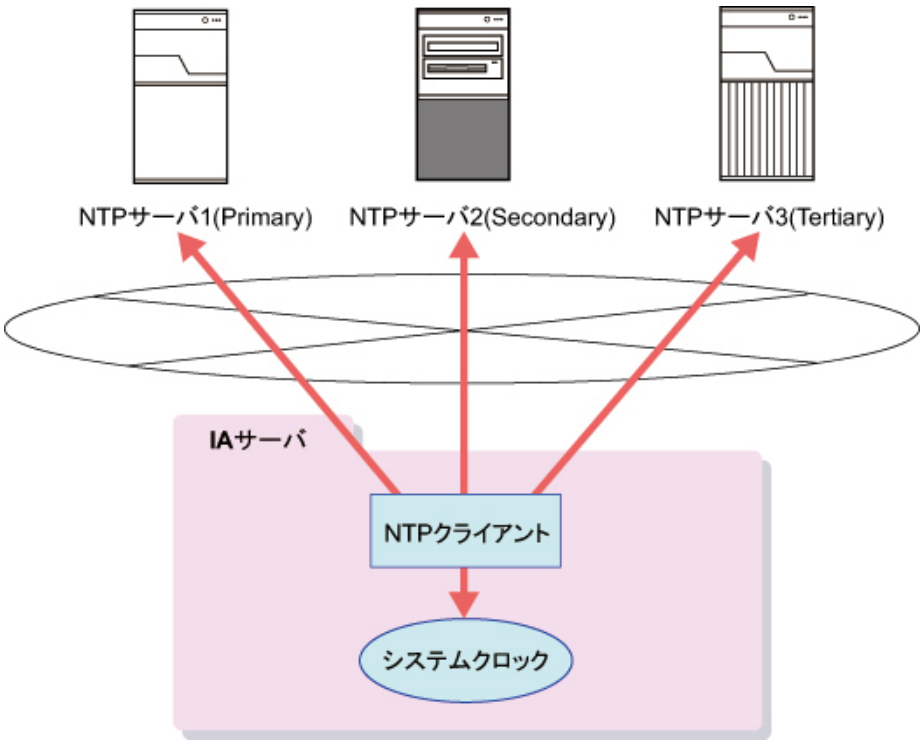
5.3.2 参照するNTPサーバの構成・参照設計

時刻同期の対象となるNTPサーバの構成について、必要となる要件を説明します。

■ 複数のNTPサーバ

安定したNTP運用を行うために、NTPクライアントから参照する外部のNTPサーバを3サーバ以上選定します。

図 32 : 複数のNTPサーバを使用する場合の概念図



ntpdは、NTPサーバとの距離、応答レスポンスの安定性など、様々な要因を考慮して選び出したNTPサーバと時刻同期します。これにより、時刻同期中のNTPサーバとの時刻同期に失敗したとしても、すぐに代わりのNTPサーバを選びだし、安定した時刻同期状態を保つことができます。NTPサーバを3サーバ以上指定することで、1サーバの時刻情報が異常な値を示しても、そのNTPサーバを排除することができます。

NTPサーバの指定が2サーバ以下の場合、正確な時刻を維持できなくなる場合があります。

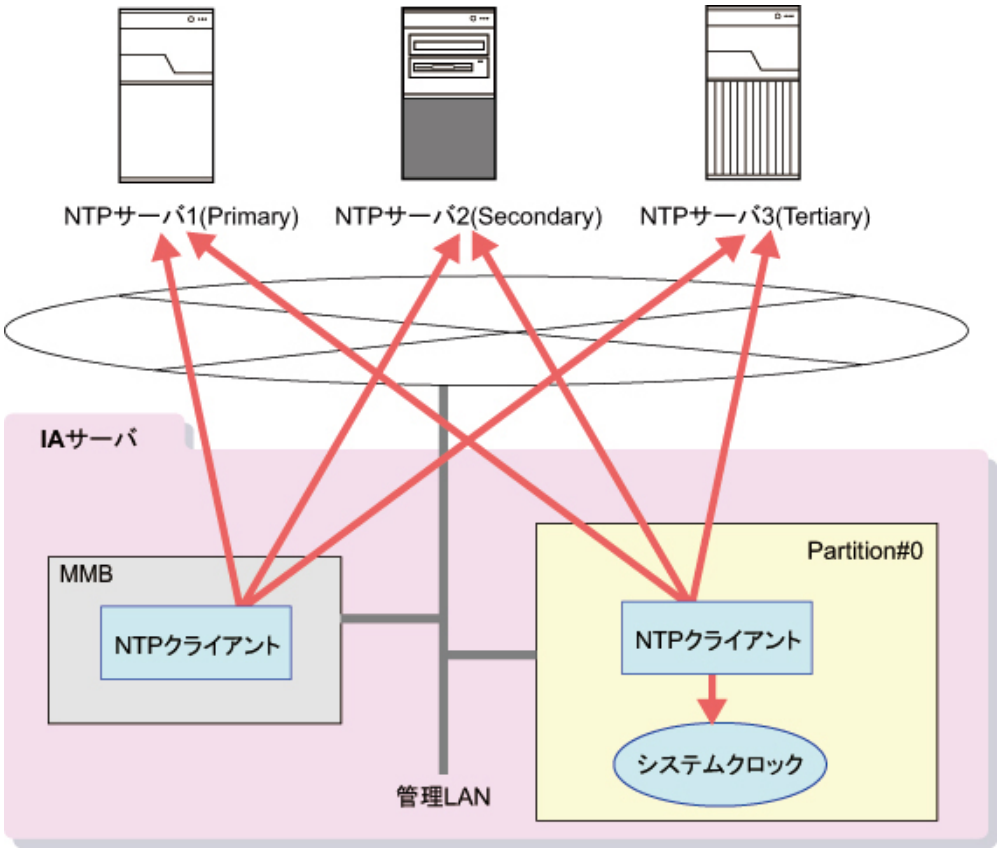
- NTPサーバが1サーバの場合
  NTPサーバとの接続が切断された場合、時刻同期するNTPサーバを失います。そのため、NTPクライアントは正確な時刻を維持できなくなります。
- NTPサーバが2サーバの場合
  2サーバのNTPサーバの時刻差が大きくなった場合、NTPクライアントはどちらのNTPサーバの時刻を使用すべきかの判断ができなくなります。そのため、NTPクライアントの維持する時刻が不安定になる場合があります。

NTPサーバは、NTP設定ファイル（/etc/ntp.conf）に記述します。設定方法については、「5.4.1 NTP設定ファイル」（→P.97）をお読みください。

■ PRIMEQUESTシリーズ固有の考慮

PRIMEQUESTシリーズでは、パーティションごとにOSを動作させることができます。それぞれのパーティションで動作させるOSでNTP運用をする場合、すべてのOSで同じNTPサーバを参照してください。同じNTPサーバを参照することにより、NTP運用が安定します。MMBでNTP運用をする場合も、同じNTPサーバを参照してください。

図 33 : PRIMEQUESTシリーズでNTPサーバを使用する場合の概念図



5.3.3 システム時計の時刻補正モード選定

ここでは、システム時計の時刻補正モードの選択について説明します。

システム時計の時刻補正モードの選択は、そのシステムで行われる業務の性格によって選択します。次の点を選択基準として検討します。

- 運用中の時刻の逆進が許容できない場合、Slewモードの選択を検討
- 正確な時刻からのずれを最小にする場合、Stepモードの選択を検討

■ Slewモード運用時の注意事項

上位NTPサーバに関する注意事項

NTPサーバをSlewモードで運用する場合には、Stepモードで運用している上位NTPサーバを参照しないでください。上位NTPサーバがStepモードで運用している場合、さらに上位のNTPサーバとの時刻差が128ミリ秒を超えた時点でStep調整が行われます。そのため、NTP運用中に上位NTPサーバとの時刻差が大きくなる可能性があります。

ntpdateサービスの運用（推奨）

Slewモードの場合、1秒間に最大0.5ミリ秒ずつシステム時計を補正するので、大きな時刻差を補正するには時間がかかります。そのため、ntpdateサービスを起動（デフォルトはStep調整）する設定を行い、ntpdをSlewモードで起動する前にStep調整によるシステム時計の時刻を補正することを推奨します。ntpdateサービスの起動設定方法については、「5.3.6 NTP環境の導入」（→P.95）をお読みください。

5.3.4 RHELシステム起動時の初期参照NTPサーバの設計

初期参照NTPサーバの設計について説明します。

ntpdateサービスを使用することによって、RHELシステム起動時にntpdateコマンド（デフォルトはStep調整）によるシステム時計の時刻を補正することができます。

ntpdateサービスの起動時は、/etc/ntp/step-tickersファイルに記述されたNTPサーバを参照するntpdateコマンドを実行します。/etc/ntp/step-tickersファイルが存在しない場合、または/etc/ntp/step-tickersファイルにNTPサーバの記述がない場合は、/etc/ntp.confに指定されたNTPサーバを参照するntpdateコマンドを実行します。詳細については、「5.4.4 時間補正関連ファイル（/etc/ntp/step-tickers）」（→P.105）をお読みください。

5.3.5 ntpdの高度な動作設計

ここでは、システム時計の時刻補正に関するntpdの高度な動作設計について説明します。

ntpdによるシステム時計の時刻補正は、参照するNTPサーバの指定とシステム時計の時刻補正モードが必須の設計項目です。さらに高度な動作が必要な場合は、それぞれの動作を設計します。

表 30：高度な動作設計の例

設計する動作内容	設定方法
• 参照するNTPサーバの優先度設定 • ローカルクロックを優先する設定（stratumの設定） • NTPクライアントのアクセス制御 • peerstats/loopstats/driftファイルの配置先変更	「5.4.1 NTP設定ファイル」（→P.97）をお読みください。 参照するNTPサーバの優先度を設定する場合は、stratum値ではなく、参照サーバのprefer値を設定してください。NTPクライアントはstratum値の小さなNTPサーバを優先して参照しますが、複数の上位NTPサーバの参照優先度を指定するためにstratum値を使用することは、本来の利用方法ではありません。
• ntpd起動時の起動オプション	「5.4.2 ntpdの起動オプション設定」（→P.103）をお読みください。
• ntpdateサービス起動時に実行されるntpdateコマンドのオプション • ntpdateサービス起動時のハードウェアクロックとの時刻同期	「5.4.3 ntpdateのオプション設定」（→P.104）をお読みください。
• drift値の設定／解除	「5.4.5 補正値記録ファイル（/var/lib/ntp/drift）」（→P.105）をお読みください。

5.3.6 NTP環境の導入

NTP環境の導入方法について説明します。

これまでのNTP環境の設計に基づき、NTP運用環境を構築します。

1. ntpdの動作状況を確認します。

```
# service ntpd status
```

ntpdサービスの動作状況を確認し、動作している場合は停止させます。

- ntpdが停止している場合

  次のメッセージが表示されます。手順2に進みます。

  ```
  ntpd is stopped
  ```

- ntpdが動作している場合

  次のメッセージが表示されますので、ntpdを停止します。

  ```
  ntpd (pid ＜PID＞) is running...
  ```

  ```
  # service ntpd stop
  ```

  ntpdが停止したことを確認するために、手順1を繰り返します。

- ntpdの停止処理中の場合

  次のメッセージが表示されますので、ntpdが停止したことを確認するために、手順1を繰り返します。

  ```
  Shutting down ntpd:                [  OK  ]
  ```

2. NTP環境設定ファイル（/etc/ntp.conf）を編集します。

これまでの設計に基づき、NTP設定ファイルを編集します。最低限の設定項目として、次の設定を行います。

a) 参照する上位NTPサーバを設定します。

ntpd設定ファイルにNTPサーバのIPアドレスを設定します。

例：NTPサーバを3サーバ指定する場合

```
server ＜信頼できるNTPサーバのIPアドレス-1＞
server ＜信頼できるNTPサーバのIPアドレス-2＞
server ＜信頼できるNTPサーバのIPアドレス-3＞
```

b) ntpdateコマンドが参照するNTPサーバを設定します。

ntpdateサービスが実行するntpdateコマンドが参照するNTPサーバのIPアドレスを/etc/ntp/step-tickersファイルに設定します。

NTP環境設定ファイルに指定したNTPサーバのIPアドレスを指定します。

```
＜信頼できるNTPサーバのIPアドレス-1＞
＜信頼できるNTPサーバのIPアドレス-2＞
＜信頼できるNTPサーバのIPアドレス-3＞
```

そのほかの設定項目については、「5.4 NTP運用の設定」（→P.97）をお読みください。

3. driftファイル（デフォルトは、/var/lib/ntp/driftファイル）の内容を確認し、値が500または-500となっている場合は、値を0に設定します。

   設定方法については、「ntpd起動前のdrift値の確認」（→P.105）をお読みください。

4. ntpdateサービスを起動します。

```
# service ntpdate start
```

   ntpdateサービスの起動が成功すると、次のメッセージが表示されます。

```
ntpdate: Synchronizing with time server:     [  OK  ]
```

5. ntpdサービスを起動します。

```
# service ntpd start ntpd
```

   ntpdサービスの起動が成功すると、次のメッセージが表示されます。

```
Starting ntpd:                [  OK  ]
```

6. 上位NTPサーバとのシステム時計の時刻同期状態を確認します。

```
# ntpq -np
```

   表示されるメッセージを確認します。IPアドレスの左端に*が表示されている行は、現在、時刻同期しているNTPサーバを示しています。

```
# ntpq -np
     remote           refid      st t when poll reach delay  offset jitter
==============================================================================
*xxx.xxx.xxx.xxx  xxx.xxx.xxx.xxx 5 u  59   64   377  0.395 -17.244 1.659
+xxx.xxx.xxx.xxx  xxx.xxx.xxx.xxx 5 u  55   64   377  0.434  3.688  3.544
+xxx.xxx.xxx.xxx  xxx.xxx.xxx.xxx 5 u  59   64   377  0.379 -24.701 6.299
```

- システム時計の時刻同期が完了するには、ntpdを起動してから5分から6分程度の時間がかかります。

7. ntpdateサービス自動起動の設定状況を確認します。

```
# chkconfig --list ntpdate
```

   次は、出力表示の一例です。

```
ntpdate   0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

   0から6までの数字は、RHELシステム起動時のランレベルを意味します。

   on/offは、そのランレベルで起動された場合のサービス自動起動の有無を表します。

   設定を変更する場合は、目的のランレベルを指定してchkconfigコマンドを実行します。

   - ntpdateサービスの自動起動を設定する場合

```
# chkconfig --level ＜ランレベル＞ ntpdate on
```

   - ntpdateサービスの自動起動を設定しない場合

```
# chkconfig --level ＜ランレベル＞ ntpdate off
```

chkconfig コマンドを実行した結果、ランレベル情報が表示されない場合、ntpdateサービスが自動起動の管理対象外になっている可能性があります。この場合、ntpdateサービスを自動起動の対象とするために、chkconfig コマンドで自動起動サービスの一覧に追加します。

```
# chkconfig --add ntpdate
```

自動実行サービスの一覧に追加した後、手順7を繰り返します。

8. RHELシステム起動時のntpdサービス自動起動の設定状況を確認します。

```
# chkconfig --list ntpd
```

次は、出力表示の一例です。

```
ntpd   0:off  1:off  2:on  3:on  4:on  5:on  6:off
```

0から6までの数字は、RHELシステム起動時のランレベルを意味します。

on/offは、そのランレベルで起動された場合のサービス自動起動の有無を表します。

設定を変更する場合は、目的のランレベルを指定して、chkconfigコマンドを実行します。

- ntpdサービスの自動起動を設定する場合

```
# chkconfig --level ＜ランレベル＞ ntpd on
```

- ntpdサービスの自動起動を設定しない場合

```
# chkconfig --level ＜ランレベル＞ ntpd off
```

chkconfig コマンドを実行した結果、ランレベル情報が表示されない場合、ntpdサービスが自動起動の管理対象外になっている可能性があります。この場合、ntpdサービスを自動起動の対象とするために、chkconfig コマンドで自動起動サービスの一覧に追加します。

```
# chkconfig --add ntpd
```

自動実行サービスの一覧に追加した後、手順7を繰り返します。

5.4 NTP運用の設定

ここでは、NTP運用を行うためのNTP設定ファイルおよびコマンドについて説明します。

5.4.1 NTP設定ファイル

/etc/ntp.confファイルは、ntpdの動作設定を行うNTP設定ファイルです。NTP設定ファイルの設定内容を変更した場合、ntpdを再起動することで設定情報をntpdの動作に反映します。

■ 参照NTPサーバ

参照する上位NTPサーバを記述します。

詳細

参照するNTPサーバのIPアドレスを記述します。複数のNTPサーバを設定できます。ローカルクロックの参照を設定する場合は、「127.127.1.0」を記述します。また、preferを記述することにより、優

先する上位NTPサーバを指定できます。ただし、上位NTPサーバとの時刻差などの要因により、設定した上位NTPサーバの参照が優先されない場合があります。

書式

```
server 〔IPアドレス｜ホスト名〕 [prefer]
```

設定例

```
server 192.168.1.2 prefer
server 0.pool.ntp.org
```

■ stratum

ローカルクロックを参照するstratumを記述します。

詳細

上位NTPサーバの代わりにローカルクロックの参照を設定する場合、次の書式で記述します。ここで記述したstratum値が、上位NTPサーバのstratumがローカルクロックよりも大きくなった場合、ローカルクロックを優先して参照します。

書式

```
fudge   127.127.1.0 stratum  階層
```

設定例

```
fudge   127.127.1.0 stratum  10
```

■ peer

同一stratum階層内のNTPサーバ間の時刻同期をするpeerを記述します。

詳細

同一stratum階層内のNTPサーバを設定します。複数のpeerを記述できます。また、preferを記述することにより、優先するpeerを指定できます。

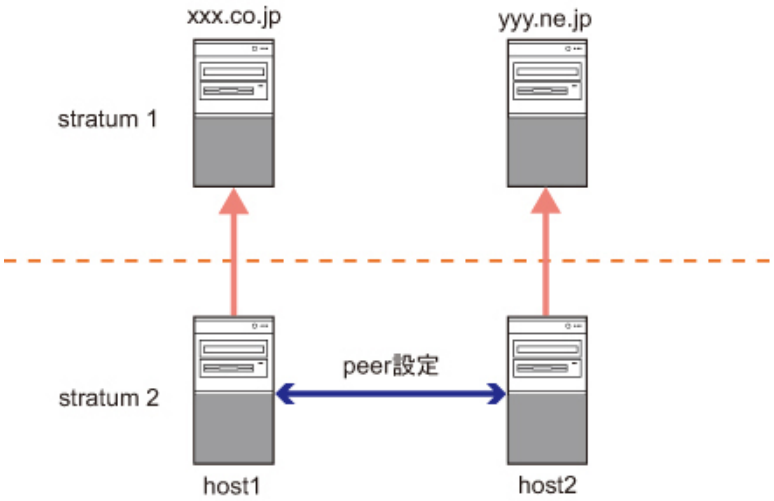
書式

```
peer   〔IPアドレス|ホスト名〕   [prefer]
```

設定例

次に、xxx.co.jpを上位stratumに持つhost1と、yyy.ne.jpを上位stratumに持つhost2をpeer設定した場合の概念図を示します。

図 **34 : peer**設定構成例



host1とhost2の/etc/ntp.confファイルの設定例は、次のとおりです。

- /etc/ntp.conf（host1）

```
server xxx.co.jp
peer   host2
```

- /etc/ntp.conf（host2）

```
server yyy.ne.jp
peer   host1
```

■ アクセス制御

ntpdに対するアクセス制御を記述します。

詳細

restrict行は、出現する順序で評価されます。そのため、最初のrestrict行は、後続のrestrict行で許可するサーバ以外からの要求を拒否するよう記述します。このとき、参照する上位NTPサーバへの許可を忘れないようにします。

書式

- IPv4アドレスによる記述

```
restrict 〔IPアドレス｜ネットワーク｜default〕 [mask サブネットマスク] [コントロール…]
```

- IPv6アドレスによる記述

```
restrict -6 〔IPv6アドレス｜ネットワーク｜default〕 [mask サブネットマスク] [コントロール…]
```

表 **31 :** ＜コントロール＞の種類

コントロール	説明
ignore	すべてのNTPに関する要求を無視します。
nomodify	時刻の問合せに応答しますが、サーバの設定変更や状態の確認／変更に使用されるパケットは無視します。

コントロール	説明
notrust	NTP認証を行っていない状態での時刻参照を許可しません。
nopeer	指定ホストと相互に時刻同期しません。
noquery	NTPの実装に依存する時刻の問合せを無視します。
notrap	リモートロギングに使用される種類のパケットに対してサービスを提供しません。
指定なし	NTPに関する要求のすべてを許可します。

設定例

```
restrict default ignore
restrict 127.0.0.1
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

■ driftファイルのパス名

driftファイルの絶対パス名を記述します。

詳細

driftファイルは絶対パス名で記述します。driftファイルの説明は、「5.4.5 補正値記録ファイル（/var/lib/ntp/drift）」（→P.105）をお読みください。

書式

```
driftfile  driftファイルの絶対パス名
```

設定例

```
driftfile /var/lib/ntp/drift
```

■ peerstats/loopstats

peerstatsファイルおよびloopstatsファイルを格納するディレクトリ名を記述します。

詳細

ディレクトリは絶対パス名で記述します。peerstatsファイルおよびloopstatsファイルの詳細については、「5.5.2 NTP運用の稼働状況の確認方法」（→P.106）をお読みください。

なお、peerstatsファイルおよびloopstatsファイルは、ntpdによる自動削除は行われませんので、定期的な監視を行い、必要に応じてファイルを削除します。ファイルを自動的に削除する手順およびスクリプト例は、「peerstats/loopstatsの設定と定期削除例」（→P.102）をお読みください。

- ファイルサイズは、NTP設定ファイルに記述するNTPサーバの数に依存します。ファイルサイズの目安は、3サーバを使用する場合、1年で約300MBです。

書式

```
statsdir  ファイル格納ディレクトリの絶対パス名
filegen ファイル生成セット名 [file ファイル名] [type タイプ] [link|nolink]
 [enable|disable]
```

statsdir ファイル格納ディレクトリの絶対パス名

ntpdの時刻同期状況を示す統計情報を格納するディレクトリ名を記述します。なお、ファイル格納ディレクトリは、ntpユーザーがアクセスできるパーミッションを設定します。

filegen ファイル生成セット名

採取する統計情報のファイル生成セットを設定します。

ファイル生成セット名を単位として、file以降の設定を 1 行で記述します。

表 **32 :** ファイル生成セット名の種類

ファイル生成セット名	説明
peerstats	参照NTPサーバとの時刻差などの統計情報を採取します。
loopstats	自サーバのシステム時計の時刻更新に関する情報を採取します。

[file ファイル名**]**

統計情報を書き込むファイルの名前を記述します。

[type タイプ**]**

統計情報を書き込むファイルの分割方法を記述します。

表 **33 :** タイプの種類

タイプ	説明	ファイル名形式
none	ファイルを分割しません。	＜ファイル名＞ 例）peerstats
pid	ntpdを起動するたびにファイルを作成します。	＜ファイル名＞.＜pid＞ 例）peerstats.9610
day	1日ごとにファイルを作成します。 UTC時刻の0時0分にファイルが作成されるため、日本時間の9時0分（+9時間）に新しいファイルが作成されます。	＜ファイル名＞.＜yyyymmdd＞ 例）peerstats.20131027
week	1週間ごとにファイルを作成します。	＜ファイル名＞.＜yyyy＞W＜num＞ 例）peerstats.2013W1
month	1か月ごとにファイルを作成します。	＜ファイル名＞.＜yyyymm＞ 例）peerstats.201310
year	1年ごとにファイルを作成します。	＜ファイル名＞.＜yyyy＞ 例）peerstats.2013
age	ntpdが24時間動作するたびにファイルを作成します。	＜ファイル名＞.a＜sec＞ 例）peerstats.a00000000

[link | nolink]

統計情報を格納するファイルを分割する場合、統計情報の書込みが行われているファイルを一意に識別するハードリンクの作成について記述します。

linkを記述すると、fileで記述したファイル名を使用してハードリンクを作成します。nolinkを記述すると、ハードリンクは作成しません。デフォルトは、linkです。

[enable | disable]

統計情報の採取について記述します。

enableを記述すると、統計情報を採取します。disableを記述すると、統計情報を採取しません。デフォルトは、enableです。

設定例

```
statsdir  /var/log/ntp/
filegen peerstats file peerstats type month link enable
filegen loopstats file loopstats type month link enable
```

■ peerstats/loopstatsの設定と定期削除例

peerstats/loopstatsの設定手順例および定期削除例について説明します。この手順は、root権限で実行します。

1. /etc/ntp.confファイルを編集します。

設定例：1か月ごとにファイルを作成し、/var/log/ntp/ディレクトリのハードリンクを作成します。

```
statsdir  /var/log/ntp/
filegen peerstats file peerstats type month link enable
filegen loopstats file loopstats type month link enable
```

2. peerstats/loopstatsファイルを格納するディレクトリを作成します。

```
# mkdir /var/log/ntp
```

3. 作成したディレクトリの所有者とグループをntpに変更します。

```
# chown ntp.ntp /var/log/ntp
```

4. ntpdを再起動します。

```
# service ntpd restart
```

5. /etc/cron.monthlyディレクトリの配下に、peerstats/loopstatsファイルを削除するシェルスクリプトを作成します。

ファイル名に日時を使用するファイル分割方法の場合を例に説明します。次は、新しい日時をファイル名に持つ6世代のファイルを保存し、それ以外のファイルを削除するスクリプト例です。

```
#!/bin/bash
STATSDIR="/var/log/ntp"    # statsdirで設定したディレクトリ
PFILE="peerstats"          # peerstatsのファイル名
LFILE="loopstats"          # loopstatsのファイル名
AGE=6                      # 保存する世代数

while [ `/bin/ls $STATSDIR/$PFILE.* | wc -l` -gt $AGE ];
do
    DELLIST=`/bin/ls $STATSDIR/$PFILE.* | /bin/sort | /usr/bin/head -n 1`
    /bin/rm -f $DELLIST
    /bin/sleep 1
done
while [ `/bin/ls $STATSDIR/$LFILE.* | wc -l` -gt $AGE ];
do
    DELLIST=`/bin/ls $STATSDIR/$LFILE.* | /bin/sort | /usr/bin/head -n 1`
    /bin/rm -f $DELLIST
    /bin/sleep 1
done
```

6. シェルスクリプトに実行権限を付与します。

```
# chmod +x ＜ファイル名＞
```

7. crondを再起動します。

```
# service crond restart
```

5.4.2 ntpdの起動オプション設定

/etc/sysconfig/ntpdファイルは、ntpd起動時のオプションを記述するファイルです。/etc/sysconfig/ntpdファイルを変更した場合、変更した設定を反映するために、ntpdを再起動します。

■ OPTIONS

ntpdの主なオプションを次に示します。

表 34 : ntpdの動作オプション

オプション	動作説明
-x	Slewモードで時刻補正を行います。
-l ファイル名	指定したファイルにログデータを書き込みます（デフォルトはシステムログ）。

書式

```
OPTIONS="ntpdの動作オプション"
```

設定例：ゆっくりとした時刻補正をする場合（**Slew**モード）

[-x]オプション（太字箇所）を追加してください。StepモードおよびSlewモードについては、「5.2.4 システム時計の補正」（→P.88）をお読みください。

```
OPTIONS="-x -u ntp:ntp -p /var/run/ntpd.pid"
```

5.4.3 ntpdateのオプション設定

/etc/sysconfig/ ntpdateファイルは、ntpdateサービス起動時に実行するntpdateコマンドへオプションを渡すためのファイルです。

■ OPTIONS

ntpdateコマンドの主なオプションを次に示します。

表 **35 : ntpdate**コマンドのオプション

オプション	動作説明
-b	参照NTPサーバとの時刻差にかかわらず、常にStepモードでシステム時計の時刻を補正します。
-B	参照NTPサーバとの時刻差にかかわらず、常にSlewモードでシステム時計の時刻を補正します。
-s	実行時のメッセージをシステムログへ書き込みます。

書式

```
OPTIONS="ntpdateコマンドのオプション"
```

設定例

```
OPTIONS="-U ntp -s -b"
```

■ SYNC_HWCLOCK

ntpdateコマンドが正常終了した際、システム時計の時刻をハードウェア時計に設定することについてのオプションです。ntpdateコマンドが異常終了した場合、この設定は参照されません。

書式

```
SYNC_HWCLOCK=｛yes｜no｝
```

yesを記述すると、システム時計の時刻をハードウェア時計に設定します。noを記述すると、システム時計の時刻をハードウェア時計に設定しません。デフォルトは、noです。

設定例

```
SYNC_HWCLOCK=yes
```

5.4.4 時間補正関連ファイル（/etc/ntp/step-tickers）

/etc/ntp/step-tickersファイルは、ntpdateサービス起動時に実行するntpdateコマンドが時刻同期を行う参照NTPサーバを記述するファイルです。

5.4.5 補正値記録ファイル（/var/lib/ntp/drift）

/var/lib/ntp/driftファイルは、参照NTPサーバと時刻同期する際に生じた時刻差などから算出された補正値をntpdが書き込むファイルです。

5.5 NTP運用の確認点

ここでは、NTP運用を開始する前に確認する事項について説明します。

5.5.1 NTP運用時の確認点

NTP運用時の確認点について説明します。

■ NTPサーバの階層を考慮したNTPサーバの起動順序

自サーバを起動する際、自サーバが参照するNTPサーバが動作していないと時刻の同期ができません。NTPサーバは、自サーバが起動する前から動作しているべきです。そのため、NTP運用をする際には、上位NTPサーバから順に起動するように調整します。

ntpdが起動してから時刻を同期することができる状態になるまで、5分から6分程度かかりますので、参照先NTPサーバを起動してから15分から20分程度（OS起動時間も含めた時間）経過した後に、自サーバを起動してください。

■ ntpdateコマンドの実施条件

ntpdateコマンドは、コマンドが実行されたときにシステム時計の時刻を補正します。しかし、ntpdateコマンド実行時にntpdが動作している場合、ntpdateコマンドはシステム時計の時刻を補正することができません。そのため、ntpdateコマンドを使用する場合（ntpdateサービスの再起動を含む）は、ntpdを停止します。

■ ntpd起動前のdrift値の確認

ntpd起動前のdrift値の確認について説明します。

ntpdを起動する前に、drift値を確認します。drift値は/var/lib/ntp/driftファイルに定期的に書き込まれています。drift値は、安定したNTP運用を行っている場合に設定されている値を維持すべきです。drift値に、その値から極端に離れた値（drift値の限界値は±500）が設定されている場合には、参照NTPサーバとの時刻同期を妨げる原因になる可能性がありますので、ntpdを起動する前にdrift値を0に変更します。実行例は次のとおりです。

drift値の設定例

1. NTP サービスを停止します。

```
# service ntpd stop
```

2. /var/lib/ntp/drift ファイルのdrift値を0に変更します。

```
# echo -n "0" > /var/lib/ntp/drift
```

3. NTP サービスを起動します。

```
# service ntpd start
```

5.5.2 NTP運用の稼働状況の確認方法

NTPサーバとの時刻同期状況、NTPサーバとの時刻差、および自サーバのシステム時計時刻更新情報の確認方法について説明します。

■ 上位NTPサーバとの時刻同期状況の確認

NTP運用を開始した際、設計どおりの時刻同期が行われていることを確認します。確認は、ntpqコマンドを使用します。

```
# ntpq -np
```

表示される各行は、時刻同期対象とするNTPサーバとの同期状況を示します。IPアドレスの左端に*が表示されている行は、現在、時刻同期しているNTPサーバを示しています。

```
# /usr/sbin/ntpq -np
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*xxx.xxx.xxx.xxx  xxx.xxx.xxx.xxx  5 u   59   64  377    0.395  -17.244  1.659
+xxx.xxx.xxx.xxx  xxx.xxx.xxx.xxx  5 u   55   64  377    0.434    3.688  3.544
+xxx.xxx.xxx.xxx  xxx.xxx.xxx.xxx  5 u   59   64  377    0.379  -24.701  6.299
(1)      (2)           (3)        (4)(5) (6)  (7) (8)    (9)     (10)    (11)
```

(1) 参照しているntpサーバとの状態を示します。

"*" ： 現在参照同期中のntpサーバ

"+" ： クロック誤り検査に合格したntpサーバ

"#" ： 時刻同期中だが、距離が遠いntpサーバ

" " ： 現在参照していないntpサーバ

"x" ： クロック誤り検査で不合格になったntpサーバ

"." ： 参照リストから外されたntpサーバ

"-" ： クラスタリング検査で参照リストから外されたntpサーバ

(2) ntpサーバのサーバ名（またはIPアドレス）

(3) ntpサーバが参照している、さらに上位のntpサーバ（不明の場合は0.0.0.0）です。".INIT."や".STEP."などと表示される場合があります。".INIT."や".STEP."は、初期化中や起動処理中に表示されます。

(4) stratum番号。参照しているntpサーバが第何階層かを表します。

(5) 階層タイプ（l:local、u:unicast、m:multicast、b:broadcast）

(6) 最後のパケットを受け取ってからの時間（単位：秒）

(7) ポーリング間隔（単位：秒）

(8) 最近8回のポーリング状況です。ポーリング状況を、成功を1、失敗を0で表すビット列を8進数で表示します。最新のポーリング状態が右端（001）になり、左にシフトするごとに過去のポーリング状態を示します。

(9) ポーリング間隔の遅延見積り（単位：ミリ秒）

(10) 階層のオフセット（単位：ミリ秒）

(11) 階層の分散（単位：ミリ秒）

■ 参照NTPサーバとの時刻差の情報確認（peerstats）

参照NTPサーバと時刻同期をした際、日時、参照NTPサーバのIPアドレス、時刻差などの統計情報を収集します。統計情報は、ntp.confに記述したディレクトリ配下に作成するファイルに書き込まれます。書き込まれた統計情報から時刻差の変化を分析することで、参照NTPサーバとの時刻の誤差を確認できます。ntpdのシステム時計の時刻補正に問題が発生した場合、この統計情報は有効な調査情報となりますので、統計情報を採取することを推奨します。NTP設定ファイルへの記述方法については、「5.4.1 NTP設定ファイル」（→P.97）をお読みください。

ファイルに書き込まれるデータ形式は、次のとおりです。

```
55112 27034.251 192.168.1.10 9614 0.000870735 0.002303132 0.018681483 0.000302981
  (1)     (2)        (3)      (4)      (5)         (6)         (7)         (8)
```

(1) 日付。修正ユリウス日を示します。

(2) 時刻（単位：秒）。0時0分0秒（UTC）からの経過秒数を示します。日本時間にするには+9時間（+32400秒）とします。

(3) 相手IPアドレス（127.127.1.0はローカルクロック）

(4) 同期先の状態（16進表示）

(5) NTPサーバとの時刻差（単位：秒）。peerstats内の時刻差は、ntpd内の内部時刻と参照NTPサーバのntpd内部時刻との差であり、OSのシステム時計と必ずしも一致していません。

(6) 遅延時間（単位：秒）

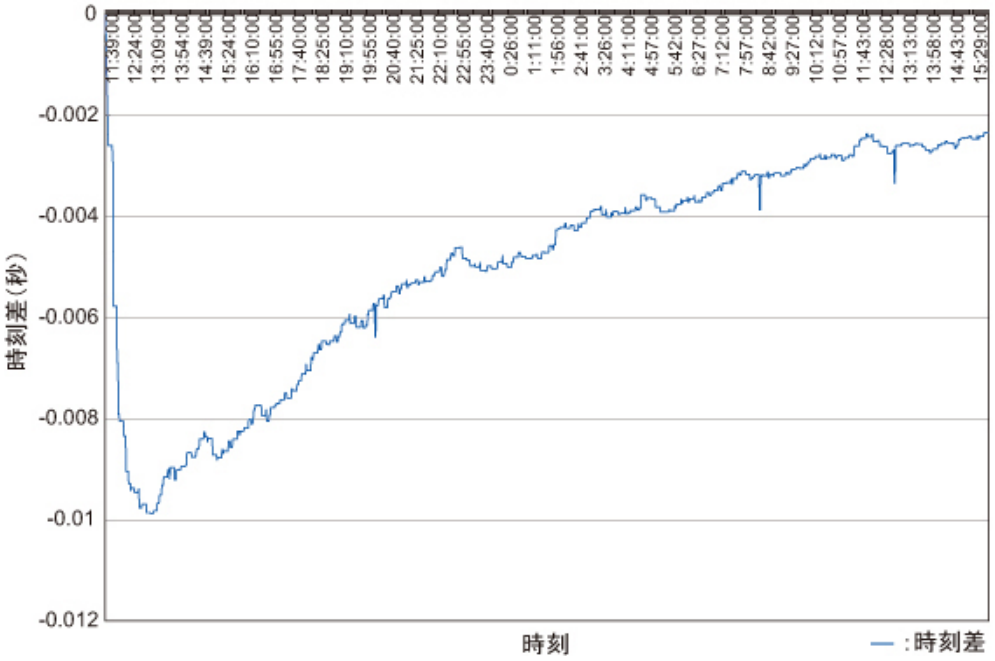
(7) 分散値（単位：秒）

(8) RMSジッタ（単位：秒）

上記出力例の場合、192.168.1.10のサーバと0.870735ミリ秒の時刻差があることを示しています。このファイルの時刻差の変化を分析することで、参照NTPサーバとどの程度の誤差で動作していたのかを検証できます。

次に、NTPサーバとの時刻差の変化の例を示します。

図 **35 : NTP**サーバとの時刻差の変化の例



■ サーバのシステム時計時刻更新に関する情報確認（loopstats）

自サーバの時刻更新に関する統計情報を収集します。統計情報は、ntp.confに記述したディレクトリ配下にファイルに書き込まれます。書き込まれた統計情報から本統計情報は、参照NTPサーバと時刻同期状態にある場合にだけ書き込まれます。ntpdの時刻補正に問題が発生した場合の調査、この統計情報は有効な調査情報となりますので、採取することを推奨します。NTP設定ファイルへの記述方法については、「5.4.1 NTP設定ファイル」（→P.97）をお読みください。

ファイルに書き込まれるデータ形式は、次のとおりです。

```
55139 21009.957 -0.000264663 -65.864 0.000095931 0.001398 6
  (1)     (2)         (3)        (4)       (5)       (6)   (7)
```

(1) 日付。修正ユリウス日を示します。

(2) 時刻（単位：秒）。0時0分0秒（UTC）からの経過秒数を示します。日本時間にするには+9時間（+32400秒）とします。

(3) NTPサーバとの時刻差（単位：秒）。loopstats内の時刻差は、ntpd内の内部時刻と参照NTPサーバのntpd内部時刻との差であり、OSのシステム時計と必ずしも一致していません。

(4) drift値（単位：PPM（parts-per-million））

(5) RMSジッタ（単位：秒）

(6) クロック安定度（単位：PPM（parts-per-million））

(7) 自ホストのクロック更新定数（polling interval）。2のベキ指数が表示されます。

5.6 うるう秒への対応

うるう秒挿入時の対応方法について説明します。

5.6.1 RHELにおけるシステム時計の調整方法

ここでは、システム時計の調整方法について説明します。
RHELのシステム時計の調整方法（時刻遷移）は、zoneinfoの使用、NTP運用の有無、および時刻補正モード（StepモードまたはSlewモード）によって異なります。

- zoneinfoディレクトリ配下のタイムゾーン定義ファイルのチェックサムを調べることで、うるう秒対応を確認できます。/etc/localtimeのチェックサム値が、次のどちらのタイムゾーン定義ファイルと一致するか確認します。

  表 **36 :** 比較するファイルとうるう秒対応状況

比較するタイムゾーン定義ファイル名	うるう秒対応状況
/usr/share/zoneinfo/Asia/Tokyo	うるう秒未対応
/usr/share/zoneinfo/right/Asia/Tokyo	うるう秒対応

  チェックサムを確認します。

```
# md5sum /etc/localtime
8470e9e89799169fde673f92103a72f8 /etc/localtime

# md5sum /usr/share/zoneinfo/Asia/Tokyo
8470e9e89799169fde673f92103a72f8 /usr/share/zoneinfo/Asia/Tokyo

# md5sum /usr/share/zoneinfo/right/Asia/Tokyo
3489e452e74de59fse645f9fj76a7sd4 /usr/share/zoneinfo/right/Asia/Tokyo
```

  この例の場合、/etc/localtimeファイルのチェックサム値は、/usr/share/zoneinfo/Asia/Tokyoのチェックサム値と一致するため、うるう秒に対応しないタイムゾーン定義ファイルを使用していることが分かります。

■ うるう秒対応のタイムゾーン定義ファイルを使用している場合

時刻遷移

NTP運用の有無にかかわらず、うるう秒挿入直時には次のように8時59分60秒を刻みます。

```
日本標準時      システム時刻
08:59:59      08:59:59
08:59:60      08:59:60
09:00:00      09:00:00
```

影響

うるう秒挿入時に8時59分60秒を刻むため、60秒を認識できないミドルウェアまたはアプリケーションに不具合（動作、表示など）が発生する可能性があります。

対処

うるう秒対応のタイムゾーン定義ファイルは使用しないでください。うるう秒対応をやめるためには、/etc/localtimeファイルをうるう秒未対応のタイムゾーン定義ファイルで上書きしてください。

実行方法

```
# cp /usr/share/zoneinfo/Asia/Tokyo /etc/localtime
```

■ うるう秒未対応のタイムゾーン定義ファイルを使用している場合（NTP運用なし）

時刻遷移

うるう秒対応のはないため、うるう秒挿入直後は次のように1秒ずれた時刻遷移になります。

```
日本標準時       システム時刻
08:59:59       08:59:59
08:59:60       09:00:00
09:00:00       09:00:01
```

影響

日本標準時と1秒ずれます。

■ うるう秒未対応のタイムゾーン定義ファイルを使用している場合（Stepモード）

時刻遷移

うるう秒挿入直時に、再度8時59分59秒を刻みます。

```
日本標準時             システム時刻
08:59:59.999       08:59:59.999
08:59:60.000       09:00:00.000
08:59:60.001       08:59:59.001          ←1秒の逆進が発生
08:59:60.002       08:59:59.002
     ～                 ～
08:59:60.999       08:59:59.999
09:00:00.000       09:00:00.000
```

影響

うるう秒挿入直時に再度8時59分59秒を刻むため、1秒の時刻の逆進が発生します。時刻が逆進することにより、ミドルウェアまたはアプリケーションの不具合が発生する可能性があります。

対処

時刻補正モードは、Slewモードを使用してください。設定方法については、「5.4.2 ntpdの起動オプション設定」（→P.103）をお読みください。なお、Slewモードによるうるう秒の対処については、次の説明をお読みください。

■ うるう秒未対応のタイムゾーン定義ファイルを使用している場合（**Slew**モード）

時刻遷移

うるう秒挿入後もそれまでどおりの時刻を刻むため、うるう秒挿入直後は1秒のずれが発生します。
しかし、その後のNTPによるシステム時計の時刻補正により、1秒のずれは徐々に解消されます。

日本標準時	システム時刻
08:59:59	08:59:59
08:59:60	09:00:00
09:00:00	09:00:01

影響

うるう秒挿入後、日本標準時と1秒ずれます。

対処

NTP運用により、1秒のずれは調整されていくため、対処は不要です。

6 システム構築：RHELシステム環境構築

ここでは、RHELシステム環境構築の方法について説明します。

6.1 RHELシステムの環境構築概要

ここでは、RHELシステムの環境構築に必要な作業概要について説明します。

6.1.1 RHELシステムの環境構築手順

ここでは、RHELシステムの環境構築の流れについて説明します。

富士通では、次の環境構築支援ツールを用意しています。

- ServerView Installation Manager（SVIM）
  ハードウェアに添付される富士通独自のインストーラです。RHELシステム、インストール作業者が選択したOSパッケージ、およびハードウェア添付ソフトウェアをインストールします。

次に、RHELシステムの環境構築の流れを示します。

「6.2 インストール前の準備」（→P.113）
サブスクリプションの登録と、インストール媒体を作成します。

「6.3 RHELのインストール」（→P.114）
SVIMを使用してインストールを行います。

「6.4 RHELシステム環境設定」（→P.115）
各種設定を手動で行います。

6.2 インストール前の準備

ここでは、RHELをインストールする前に実施する準備内容について説明します。

6.2.1 カスタマーポータルへのサブスクリプションの登録・更新

サブスクリプションの登録・更新について説明します。

RHELを使用するためには、Red Hat社の提供するカスタマーポータルにて、サブスクリプションの登録が必要です。サブスクリプションの登録をすることにより、RHELのインストールイメージファイル（ISOファイル形式）、バグ・セキュリティ修正の入手など、Red Hat社のサポートサービスを受けることができます。

サブスクリプションの登録・更新の手順については、富士通のSupportDeskご契約者様向けサイトSupportDesk-Web（http://eservice.fujitsu.com/supportdesk/）を参照してください。

6.2.2 インストール媒体の作成

RHELシステム環境の構築に必要なインストール媒体の準備について説明します。

■ インストールDVDの作成

インストールDVDの作成手順を説明します。

カスタマーポータルからインストールするRHELのリリースに対応するインストールイメージファイル（ISOイメージ）をダウンロードして、インストールDVDを作成します。

- 次の場合は、インストールDVDの作成は不要です。
  - リモートストレージ機能でISOイメージファイルを使用する場合

1. カスタマーポータル（https://access.redhat.com）にログインします。

2. ISOイメージファイルの公開サイトページを開きます。
   インストールするRHELリリースを選択します。

- ここでは最新のリリースが表示されます。使用するISOイメージファイルが表示されない場合、「以前のリリースのISOイメージの表示」をクリックして目的のRHELリリースを表示します。

3. ISOイメージファイルをダウンロードします。
   カスタマーポータルの画面にmd5チェックサムが表示されていますので、ダウンロードしたISOイメージファイルのチェックサムと照合します。

4. ダウンロードしたISOイメージファイルをイメージ形式でDVDに書き込み、インストールDVDを作成します。

注意

- 作成したインストールDVDを参照すると、書き込んだファイル名が表示される場合は、データ形式でDVDに書き込んでいます。イメージ形式でDVDを再作成してください。

参考

- 作成したインストールDVDは、RHELシステムをレスキューモードで起動する場合にも使用します。

6.3 RHELのインストール

ここでは、RHELをサーバ（PRIMEQUESTシリーズ）にインストールする方法について説明します。

6.3.1 RHELのインストールにおける注意事項

RHELのインストール時の注意事項について説明します。

■ SAN Boot環境の設定

SAN Boot環境を構築するためには、RHELのインストール前にサーバの環境設定が必要です。機種ごとに対応方法が異なりますので、次の対応を実施してください。

- PRIMEQUESTシリーズ
  『導入マニュアル』に従って、サーバ環境を設定します。

参考

- SAN Boot環境とは、筐体に内蔵されたディスクではなく、ファイバーチャネルで接続した外部のSANストレージ装置のディスクからシステムを起動する環境です。

■ SAN環境におけるマルチパス構成時の考慮

SAN環境（SAN Boot環境を含む）をマルチパス構成とする場合は、RHELのインストールおよびマルチパスドライバの導入が完了するまでは、次の構成を維持してRHELシステムの環境を構築します。

- RHELのインストールの対象となる1つのディスクアレイ装置、LUNマッピング、Affinity Group、およびゾーンだけを接続する。
  複数台のディスクアレイ装置、複数のLUNマッピング、Affinity Group、およびゾーンは、RHELシステム環境の構築後に接続します。
- ディスクアレイ装置との接続は、1系統（＝1パス）だけとする。
  マルチパス接続されている状態でRHELをインストールした場合、マルチパス接続の影響により、1つのディスク装置が複数のディスクとして認識されてしまい、正常な環境構築ができません。

6.3.2 RHELのインストール方法

RHELのインストール方法について説明します。

RHELをPRIMEQUESTシリーズにインストールするには、ハードウェアに添付されるSVIMを使用します。

SVIMを使用したインストールを説明したドキュメントは、次表のとおりです。ドキュメントをお読みになり、SVIMを実行するために必要な作業手順の確認およびSVIMで必要となる入力項目を決定します。

表 37 : RHELインストール時の参照ドキュメント

機種	ドキュメント名
PRIMEQUESTシリーズ	『導入マニュアル』（ハードウェア添付） 『ServerView Installation Manager』（ハードウェア添付） 『Linux ユーザーズマニュアル』（本書）

次に、SVIMを使用するうえでの考慮点を説明します。

- ディスクパーティションを設定する場合、swapパーティション（ファームウェアがUEFIモードの場合、/boot/efiパーティションおよびswapパーティション）を除いたパーティションは、ext3ファイルシステムにすることを推奨します。詳細は、「1.2.2 ファイルシステム使用上の注意事項」（→P.19）をお読みください。
- インストールモードのクイックモードを使用すると、ext4ファイルシステムが自動選択されます。これ以外のファイルシステム種別を選択する場合、ガイドモードを使用してください。
- インストールの過程で表示される基本設定画面では、言語をJapaneseに変更してください。

■ インストール後の作業

SVIM を使用してインストールした後は、「6.4 RHELシステム環境設定」（→P.115）を参照し、必要な環境設定を行ってください。

6.4 RHELシステム環境設定

ここでは、RHELシステムの環境設定について説明します。
RHELシステムの環境設定は、RHELのインストールが完了したあとに実施します。

6.4.1 SAN環境におけるマルチパス構成

SAN環境（SAN Bootを含む）をマルチパス構成とする場合の対処について説明します。

1. マルチパス用ドライバをインストールします。
   詳細については、ドライバソフトに添付されるドキュメントをお読みください。

2. RHELシステムを停止します。

3. 未接続のディスクアレイ装置を接続します。

複数台のディスクアレイ装置、複数のLUNマッピング、Affinity Group、およびゾーンと接続する場合、マルチパス構成も含めて、この手順を実行するときにすべての装置を接続します。

4. RHELシステムを起動し、マルチパス用ドライバの設定を行います。

   詳細については、ドライバソフトに添付されるドキュメントをお読みください。

6.4.2 デバイス名の変更防止

ハードウェア故障などによるデバイス名ずれの防止設定を実施します。

設定方法については、「3.3 ディスク系デバイスにおける対処」（→P.50）をお読みください。

6.4.3 時刻補正（NTP）

NTPによる時刻補正の設定を実施します。

設定方法については、「5 システム設計：時刻同期（NTP）」（→P.82）をお読みください。

6.4.4 ファイルシステムオプション

ファイルシステムのオプションを指定して利用する場合、かつRHELシステム起動時から有効にする場合は、/etc/fstabファイルに利用するファイルシステムのオプションを指定します。

/etc/fstabファイルは、RHELシステム起動時にマウントするファイルシステムに関する情報を記載する設定ファイルです。富士通が推奨するオプションについては、「1.2.3 ファイルシステムのマウントオプション」（→P.20）をお読みください。

1. 編集前の/etc/fstabファイルを別ファイル名で退避します。

```
# cp -p  /etc/fstab  /etc/fstab.org
```

2. /etc/fstabファイルを編集・保存します。

   編集行内の4番目のフィールドに、カンマ区切りでオプションを記述します。オプションの設定が必要なすべてのファイルシステムに対して、編集を行います。

   例：ext3形式で作成した/homeファイルシステムに対して、[data_err]オプションを追加する場合

```
UUID=＜ファイルシステムのUUID＞  /home  ext3  defaults,data_err=abort  1  1
```

3. RHELシステムを再起動します。

```
# shutdown -r now
```

6.4.5 OOM Killerの動作

システムのメモリ使用量が増加し、メモリ枯渇状態になった場合のシステムの動作を設定します。

■ OOM Killerとは

OOM Killerは、システムのメモリ使用量が増加し、メモリ枯渇状態（Out Of Memory）になった場合に、不特定プロセスを強制終了することでメモリを解放する機能です。

■ OOM Killerに関するカーネルパラメーターの設定

メモリ枯渇状態になった場合のシステムの動作は、vm.panic_on_oomパラメーターで指定します。このパラメーターにはメモリ枯渇状態になった場合に、OOM Killerを動作させるか、またはシステムパニックさせるかを設定します。

システムを再起動するまでの一時的な設定の場合と恒久的な設定の場合で、設定方法が異なります。

- 一時的な設定の場合
  sysctlコマンドのvm.panic_on_oomパラメーターで設定します。
- 恒久的な設定の場合
  /etc/sysctl.confファイルのvm.panic_on_oom行で設定します。

次表に、vm.panic_on_oomパラメーターの設定値について示します。

表 38 : vm.panic_on_oomパラメーターの設定値と動作

設定値	動作説明
0（デフォルト）	OOM Killerを動作させます。
1	次のように動作させます。 • システム全体がメモリ枯渇状態のとき、システムパニックさせます。 • CPUSETS機能またはmempolicy機能でメモリ量を設定したプロセスが設定メモリ量を超えたとき、OOM Killerを動作させます。
2	OOM Killerを動作させず、システムパニックさせます。

- CPUSETS機能については、Red Hat 社のマニュアル『Resource Management Guide』をお読みください。
- mempolicy機能については、オンラインマニュアルで次の項目を参照してください。
  mbind(2)、set_mempolicy(2)、get_mempolicy(2)、numa(7)

■ OOM Killer抑止機能

OOM Killerの強制終了から除外する場合、プロセスごとの/proc/＜PID＞/oom_adjファイルに「-17」を設定します。「-17」以外を設定すると、OOM Killerの強制終了から除外できません。

RHEL6.2以降では、プロセスごとの/proc/＜PID＞/oom_adjファイルに「-17」を設定、またはプロセスごとの/proc/<PID>/oom_score_adjファイルに「-1000」を設定することで、OOM Killerの強制終了から除外することができます。

なお、＜PID＞は、除外するプロセスのプロセスIDです。

6.5 レスキューモードの起動（トラブル発生時）

RHELマイナーリリースに対応するインストール媒体を使用して、最小限の機能のRHELシステムを起動することをレスキューモードと呼びます。

レスキューモードは、主に次の場合に使用します。

- システムディスクのRHELシステムが起動しない場合
- システムディスクを使わずにRHELシステムを起動する必要がある場合（ファイルシステム修復など）

- システムディスクをオフラインでバックアップまたはリストアする場合

- レスキューモードで接続済みデバイスを認識しない場合、後述の手順で、ドライバを組み込んでください。

6.5.1 レスキューモード起動準備

レスキューモードを使用する際に、事前に準備しておく事項について説明します。

■ RHELのインストール媒体の準備

RHELシステムをレスキューモードで起動するため、RHELインストールの際に使用したインストール媒体を用意します。

■ 起動デバイスの優先順位の設定

起動デバイスの優先順位を変更し、RHELインストール媒体が起動デバイスに選ばれるようにサーバを設定します。起動デバイスの優先順位の設定方法は、サーバのマニュアルをお読みください。

6.5.2 レスキューモードの起動

レスキューモードの起動方法について説明します。

1. インストール媒体をサーバに挿入して、サーバの電源を投入します。

2. 「Welcome」画面で［Rescue installed system］を選択します。

3. 「Choose a Language」画面で［English］を選択します。
   ここで［Japanese］を選択すると、LANG=ja_JP.UTF-8の環境となりますが、一部のコマンドで画面表示が乱れることがあるため、必ず［English］を選択します。

4. 「Keyboard Type」画面で［jp106］を選択します。
   フラットディスプレイ（PG-R1DP2）などの付属キーボードを使用する場合には、［us］を選択します。

5. 「Rescue Method」画面で「Local CD/DVD」を選択します。
   サーバによっては、本画面が表示されない場合があります。表示されない場合は、手順8以降を実施してください。

6. 「No driver found」画面で「Select driver」を選択します。
   サーバによっては、本画面が表示されない場合があります。表示されない場合は、手順8以降を実施してください。

7. 「Setup Networking」画面でネットワーク環境を設定するか選択します。
   ネットワークを利用しない場合は「No」を選択します。

8. 「Rescue」画面で既存のパーティションを自動的にマウントするかどうかのメッセージが表示されますので、［Skip］を選択します。

9. 「First Aid Kit quickstart menu」画面で［shell Start shell］を選択します。
   コマンドプロンプトが表示されます。

10. ディスクにアクセスする場合、ディスクをマウントします。
    例：/dev/sda2をマウントする場合

```
# mkdir /mnt/mountpoint
# mount /dev/sda2 /mnt/mountpoint
```

6.5.3 レスキューモードの終了方法

レスキューモードを終了する場合は、インストール媒体をサーバから取り出したあと、コマンドプロンプトでrebootコマンドを実行します。
これにより、レスキューモードで起動したRHELシステムが終了し、システムディスクのRHELシステムが起動します。

```
# reboot
```

7 運用・保守：バックアップ・リストア

ここでは、バックアップ・リストアについて説明します。

7.1 バックアップ・リストアの概要

バックアップ・リストアの概要について説明します。

7.1.1 バックアップ・リストアの要件

システム資産の保全には、バックアップおよびリストア計画を立案します。
バックアップ・リストアは、次の要件を満たすことが重要です。

- 意図した状態に復元できること（確実性、安全性）
- 業務への影響を最小限に抑止すること（スピード）

- バックアップしたデータは、バックアップしたRHELシステムにだけリストアします。特に、システムディスクの場合、異なる機種間および同一機種でもハードウェア構成が異なる場合は、RHELシステムが起動できなくなる可能性があるので、別RHELシステムのデータのリストアは絶対にしないでください。
- 異なるRHELリリースで実施した場合、restoreコマンドが失敗する可能性があるので、dumpコマンドとrestoreコマンドは、同じRHELリリースで実行します。restoreコマンドの実行に備えて、dumpコマンド実行時のRHELリリースに対応するインストール媒体を用意します。

7.1.2 バックアップの方法

バックアップの形態には、オンラインバックアップとオフラインバックアップがあります。

■ オンラインバックアップ

オンラインバックアップは、RHELシステム、業務アプリケーションなどを停止せずにバックアップを実行します。このため、業務アプリケーションなどに関連するファイル間の整合性を保つために、特別な機能を持つオンラインバックアップ対応製品を使用します。なお、本書では、オンラインバックアップについては説明しません。

■ オフラインバックアップ

オフラインバックアップは、業務アプリケーションなどを停止させ、かつRHELシステムをバックアップする必要最低限の状態にしてから、バックアップを実行します。バックアップの運用手順は簡単ですが、常駐デーモンの停止、常駐プロセスの停止、および業務アプリケーションの停止が必要です。業務停止時間はバックアップするディスク領域の容量に比例し、バックアップ対象のディスク領域が大きいほど業務停止時間は長くなります。

オフラインバックアップには、次の3つの形態があります。

- フルバックアップ
  バックアップ対象領域すべてをバックアップします。リストア時には、リストア対象日のバックアップデータを使用します。
- 差分バックアップ
  フルバックアップ時点からの変更データをバックアップとして採取します。リストア時には、最新のフルバックアップデータとリストア対象日の差分バックアップデータを使用します。
- 増分バックアップ
  直前に実施した増分バックアップ時点からの変更データをバックアップとして採取します。リストア時には、最新のフルバックアップデータとリストア対象日までのすべての増分バックアップデータを使用します。

図 36 : バックアップ形態

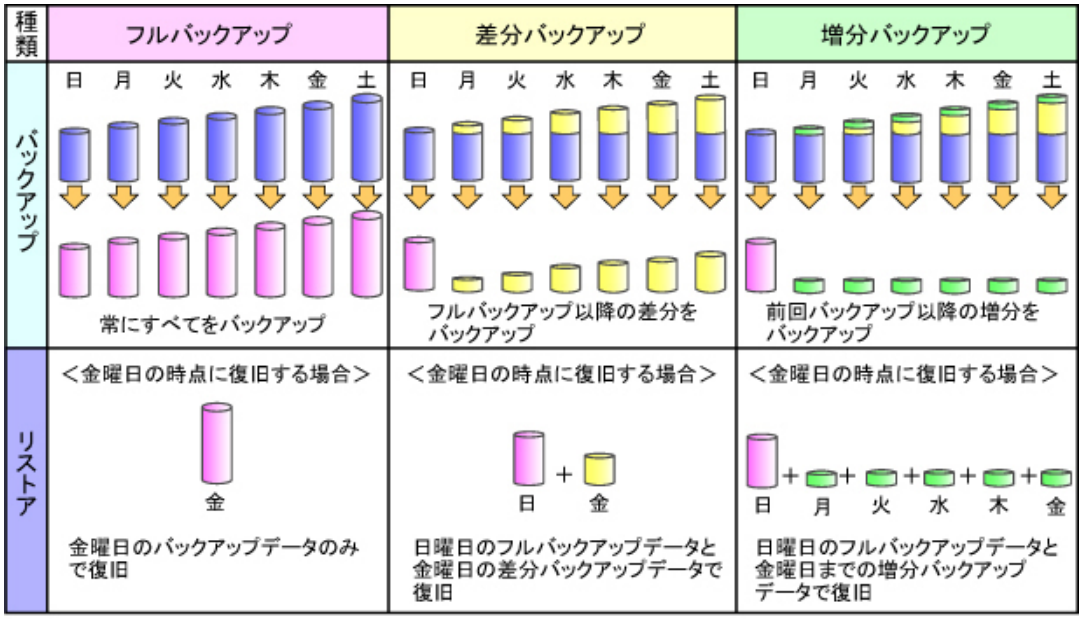


表 39 : バックアップ形態の比較

項目	フルバックアップ	差分バックアップ	増分バックアップ
バックアップサイズ	大きい	小さい	小さい
バックアップ時間	長い	～中	短い
フルリストア時間	短い	中～	長い
管理のしやすさ	容易	やや複雑	複雑

7.1.3 バックアップデバイス

ここでは、バックアップデータを保存するデバイスの構成について説明します。
バックアップデータは、ローカル接続デバイスまたはリモート接続デバイスに保存します。なお、ローカル接続デバイスとリモート接続デバイスでは、バックアップ・リストアの実行方法が異なります。そのため、本書ではバックアップ・リストアを次のように表記して、両者を区別します。

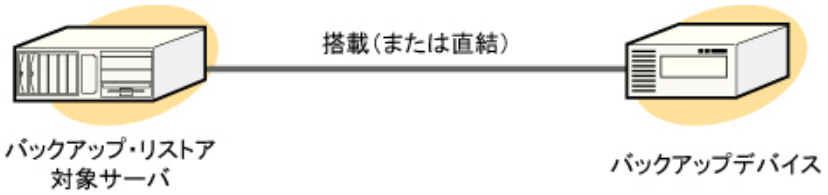
表 **40 :** バックアップデバイス種別による操作表記

表記	説明
ローカルバックアップ	ローカル接続デバイスを使用したバックアップ
ローカルリストア	ローカル接続デバイスを使用したリストア
リモートバックアップ	リモート接続デバイスを使用したバックアップ
リモートリストア	リモート接続デバイスを使用したリストア

■ ローカルバックアップ、ローカルリストア

ローカルバックアップおよびローカルリストアは、サーバに接続されたデバイスを使用します。

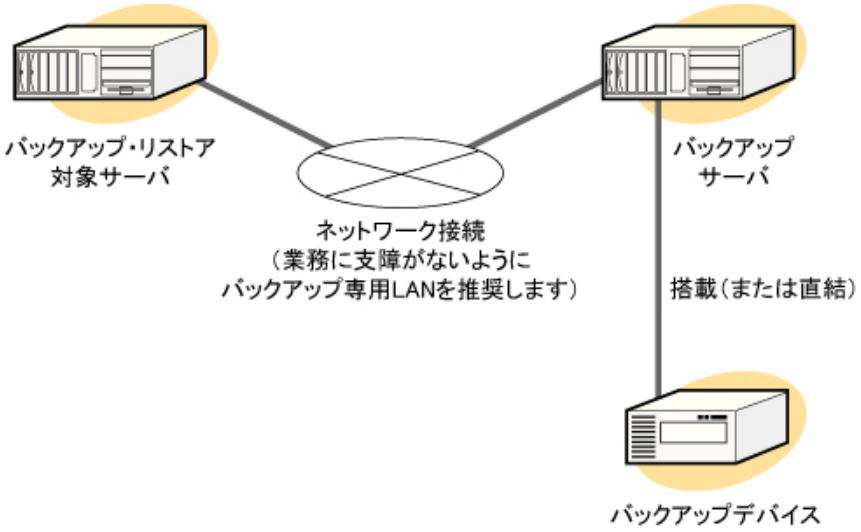
図 **37 :** ローカルバックアップ、ローカルリストアのデバイス



■ リモートバックアップ、リモートリストア

リモートバックアップおよびリモートリストアは、ネットワーク経由でバックアップサーバに接続されたデバイスを使用します。

図 **38 :** リモートバックアップ、リモートリストアのデバイス



注 意

- 次の環境では使用しないでください。
  - インターネットを経由した環境（VPNなどを使用してセキュアな通信環境が確保されている場合を除く）
  - NAT/NAPTを利用した環境
  - ファイアーウォールやプロキシを経由した環境

7.1.4 バックアップ前の確認事項

ここでは、バックアップ・リストアのための準備について説明します。

■ コマンドのインストール確認

rpmコマンドを使用して、バックアップ・リストアで使用するコマンドがインストールされていることを確認します。インストールされていない場合は、バックアップを実行する前に次のコマンドをインストールします。

表 41 : 確認するコマンド

コマンド	パッケージ名	説明
mt	mt-st	テープの操作
dump	dump	バックアップに関する操作
restore	dump	リストアに関する操作
tar	tar	バックアップ・リストアに関する操作

■ ファイルシステム情報の取得

バックアップは、ファイルシステムをバックアップ単位とします。そのため、ディスクパーティションごとのファイルシステム種別を明確にしておきます。

リストアは、バックアップデータをリストアするファイルシステムのマウントが必要です。また、ディスク交換した場合は、ディスクパーティションの再構築が必要になります。そのため、ディスクパーティションの詳細情報を取得しておきます。

1. 自動マウントするファイルシステム情報を取得します。

例：/etc/fstabファイルの場合

```
UUID=＜/パーティションのUUID＞     /         ext3 defaults   1 1
UUID=＜/bootパーティションのUUID＞ /boot     ext3 defaults   1 2
UUID=＜swapパーティションのUUID＞   swap     swap defaults   0 0
tmpfs        /dev/shm           tmpfs    defaults        0 0
devpts       /dev/pts           devpts   gid=5,mode=620  0 0
sysfs        /sys               sysfs    defaults        0 0
proc         /proc              proc     defaults        0 0
```

例：/etc/fstab（ファームウェアがUEFIモードの場合）

/boot/efiにvfatファイルシステムが使用される設定が表示されます。

```
UUID=＜/パーティションのUUID＞       /           ext3     defaults       1 1
UUID=＜/boot/efiのボリュームID＞     /boot/efi  vfat
umask=0077,shortname=winnt 0 0
UUID=＜swapのUUID＞    swap                swap       defaults       0 0
tmpfs                /dev/shm            tmpfs      defaults       0 0
devpts               /dev/pts            devpts     gid=5,mode=620 0 0
sysfs                /sys                sysfs      defaults       0 0
proc                 /proc               proc       defaults       0 0
```

2. すべてのディスクパーティション情報を取得します。

例：fdiskコマンドによる確認（デバイスが/dev/sdaで、パーティションテーブルがMS-DOS形式の場合）

```
# fdisk -u -c /dev/sda

コマンド (mでヘルプ): p

ディスク /dev/sda: 73.4 GB, 73407868928 バイト
ヘッド 255, セクタ 63, シリンダ 8924, 合計 143374744 セクタ
Units = セクタ数 of 1 * 512 = 512 バイト
セクタサイズ (論理 / 物理): 512 バイト / 512 バイト
I/O size (minimum/optimal): 512 bytes / 512 bytes
ディスク識別子: ＜0xから始まるディスク識別子＞

デバイス ブート        始点        終点   ブロック     Id  システム
/dev/sda1            2048    526335  262144    83   Linux
/dev/sda2          526336  84330495  41902080  83   Linux
/dev/sda3       84330496  88524799  2097152   83   Linux
```

例：partedコマンドによる確認（パーティションテーブルがGPT形式の場合）

```
# parted /dev/sda

(parted) unit s
(parted) print

モデル: FUJITSU MBD2147RC (scsi)
ディスク /dev/sda: 286749488s
セクタサイズ (論理/物理): 512B/512B
パーティションテーブル: gpt

番号     開始        終了          サイズ       タイプ     ファイルシステム   フラグ
1     2048s       9764863s    9762816s    primary    fat32            boot
2     9764864s    39061503s   29296640s   primary    ext3
3     39061504s   49301503s   10240000s   primary    linux-swap(v1)
(parted) quit
```

3. すべてのファイルシステムのUUIDを取得します。

```
# blkid /dev/sda1
/dev/sda1: UUID="＜/dev/sda1のUUID＞" TYPE="ext3"
# blkid /dev/sda2
/dev/sda2: UUID="＜/dev/sda2のUUID＞" TYPE="ext3"
```

4. ラベル名を使用している場合、すべてのファイルシステムのラベル名を取得します。

```
# e2label /dev/sda1
/boot
# e2label /dev/sda2
/
```

swapのディスクパーティションに設定されているラベル名を確認するには、次のコマンドを実行して、TYPE="swap"のデバイスを表示させます。

```
# blkid | grep swap
/dev/sda3: UUID="＜swapのUUID＞" TYPE="swap" LABEL="SWAP-sda3"
```

7.2 バックアップ前の準備

バックアップを実行する前に必要な操作について説明します。

7.2.1 ランレベル1（シングルユーザーモード）への移行

バックアップを実行するために、RHELシステムをバックアップする必要最低限の状態にします。

ランレベル 1 に移行する手順を説明します。

この手順では、ブートローダの設定ファイルとして、/boot/grub/grub.conf（ファームウェアがBIOSまたはUEFIのBIOS互換モードの場合に使用される設定ファイル）を使用して説明します。ファームウェアがUEFIモードの場合の設定ファイルは、/boot/efi/EFI/redhat/grub.confですので、適宜、読み替えてください。

1. /boot/grub/grub.confファイルの編集を開始します。

```
# vi /boot/grub/grub.conf
```

2. カーネル版数およびファイルシステムのラベル名をランレベル1（シングルユーザーモード）の起動エントリーに追加します。

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
password --md5 ＜ユニークな文字列＞
title Red Hat Enterprise Linux (2.6.32-71.el6.x86_64)
          root (hd0,0)
          kernel /vmlinuz-2.6.32-71.el6.x86_64 ro root=UUID=＜/パーティションのUUID＞ rd_NO_LUKS rd_NO_LVM rd_NO_MD rd_NO_DM LANG=ja_JP.UTF-8
KEYBOARDTYPE=pc KEYTABLE=jp106 rhgb quiet
          initrd /initramfs-2.6.32-71.el6.x86_64.img
title Red Hat Enterprise Linux (2.6.32-71.el6.x86_64) Single User Mode
          root (hd0,0)
          kernel /vmlinuz-2.6.32-71.el6.x86_64 ro root=UUID=＜/パーティションのUUID＞ rd_NO_LUKS rd_NO_LVM rd_NO_MD rd_NO_DM LANG=ja_JP.UTF-8
KEYBOARDTYPE=pc KEYTABLE=jp106 rhgb quiet 1  ←「1」を指定
          initrd /initramfs-2.6.32-71.el6.x86_64.img
```

追加したエントリーのkernel行の末尾に、「1」を指定します。これは、ランレベル1（シングルユーザーモード）でRHELシステムを起動することを示します。

3. RHELシステムを再起動します。

```
# shutdown -r now
```

4. グラフィカルブートローダ(GRUB)画面で追加したエントリーを選択し、ランレベル1（シングルユーザーモード）に移行します。

- システムをランレベル1（シングルユーザーモード）に移行するために、initコマンドは使用しないでください。initコマンドを使用した場合、サービスが完全に停止しない可能性があります。この場合、バックアップするディスクパーティションを占有できないため、バックアップを実行できません。

7.2.2 ファイルシステムの整合性チェック

ファイルシステムの整合性をチェックし、正常な状態であることを確認します。

■ ext3ファイルシステムまたはext4ファイルシステムの場合

マウントポイント/bootにマウントしている/dev/sda1ディスクパーティション（ext3ファイルシステムまたはext4ファイルシステム）を例に説明します。

1. /bootファイルシステムをアンマウントします。

```
# cd /
# umount /boot
```

- umountコマンドがエラーとなった場合、手順2のe2fsckコマンドを実行しないでください。エラーは、ファイルシステムが何らかの原因で使用中である可能性があります。原因を特定して、ファイルシステムが使用されていない状況にしてから、アンマウントしてください。

2. アンマウントしたディスクパーティションのファイルシステムをチェックします。

```
# e2fsck -nfv /dev/sda1
```

ファイルシステムのエラーを検出した場合、RHELシステムをレスキューモードで起動し、該当ディスクパーティションのファイルシステムを復旧します。レスキューモードの起動については、「6.5 レスキューモードの起動（トラブル発生時）」（→P.117）をお読みください。

■ vfatファイルシステムの場合

マウントポイント/boot/efiにマウントしている/dev/sda1ディスクパーティション（vfatファイルシステム）を例に説明します。

- 本手順は、ファームウェアがUEFIモードの場合に該当します。

1. /boot/efiファイルシステムをアンマウントします。

```
# cd /
# umount /boot/efi
```

- umountコマンドがエラーとなった場合、手順2のfsckコマンドを実行しないでください。エラーは、ファイルシステムが何らかの原因で使用中である可能性があります。原因を特定して、ファイルシステムが使用されていない状況にしてから、アンマウントしてください。

2. アンマウントしたディスクパーティションのファイルシステムをチェックします。

```
# fsck -nfv /dev/sda1
```

ファイルシステムのエラーを検出した場合、RHELシステムをレスキューモードで起動し、該当ディスクパーティションのファイルシステムを復旧します。レスキューモードの起動については、「6.5 レスキューモードの起動（トラブル発生時）」（→P.117）をお読みください。

■ /（ルート）などアンマウント不可能なファイルシステムの場合

/（ルート）にマウントしている/dev/sda2ディスクパーティションを例に説明します。

1. /（ルート）ファイルシステムを読込み専用で再マウントします。

```
# cd /
# mount -r -n -o remount /
```

- 読込み専用のマウントがエラーとなった場合、手順2のe2fsckコマンドを実行しないでください。ファイルシステムがマウント状態のままe2fsckコマンドを実行すると、チェックしているディスクパーティションのファイルシステムを破壊する可能性があります。「7.2.1 ランレベル1（シングルユーザーモード）への移行」（→P.125）に沿って、ランレベル1（シングルユーザーモード）でRHELシステムを再起動してから、再マウントしてください。

2. 読込み専用で再マウントしたディスクパーティションのファイルシステムをチェックします。

```
# e2fsck -nfv /dev/sda2
```

ファイルシステムのエラーを検出した場合、ディストリビューションのブータブル媒体からRHELシステムをレスキューモードで起動し、該当ディスクパーティションのファイルシステムを復旧します。レスキューモードの起動については、「6.5 レスキューモードの起動（トラブル発生時）」（→P.117）をお読みください。

3. チェックしたディスクパーティションのファイルシステムを元のマウント状態に戻します。

```
# mount -w -n -o remount /
```

7.3 バックアップの実施

ここでは、バックアップの手順について説明します。
バックアップは、ローカルバックアップ、リモートバックアップ、およびバックアップ媒体の種類（テープ装置またはデータカートリッジ）によって手順が異なります。環境に応じたバックアップ手順を確認します。

- バックアップのスケジューリング、レポート機能、グラフィカル画面での操作など、RHEL標準機能ではこれらの高度な機能には対応できません。より高度な機能を必要とする場合、NetVaultの使用を推奨します。
- mtコマンドおよびdumpコマンドに指定するテープ装置（DAT、LTO）の互換デバイス名は、コマンド終了時のテープ巻戻しの扱いにより、次の2種類を使い分けます。
  - /dev/st0
    コマンド終了時に、テープを巻き戻す場合に使用します。
  - /dev/nst0
    コマンド終了時に、テープを巻き戻さずに、そのままにする場合に使用します。

7.3.1 テープ装置を使用したローカルバックアップ

ローカル接続のテープ装置（DAT、LTO）を使用して、ディスクパーティションをバックアップする手順を説明します。

- バックアップとリストアをそれぞれ異なるテープ装置で実行する場合、テープ装置のデフォルトのブロックサイズが異なる場合があります。バックアップ時のブロックサイズとリストア時のブロックサイズが異なると、リストアできませんので、バックアップとリストアのブロックサイズを一致させてください。
- テープ装置が他のサーバと共有されているなどの理由で、他者によってテープ装置のブロックサイズが変更されていることがあります。その場合は、テープ装置が他のサーバで利用されていない状態で、ブロックサイズを設定してください。
  例：可変長に設定する場合

```
# mt -f /dev/nst0 setblk 0
```

  例：固定長10240byteに設定する場合

```
# mt -f /dev/nst0 setblk 10240
```

- テープ装置が他のサーバと共有されているなどの理由でテープ装置がロックされ、使用できない場合があります。 このときは、ロックされた原因を特定し、使用可能な状態にしてください。
- テープ装置のブロックサイズを固定長に設定する場合は、バックアップに使用するdumpコマンドまたはtarコマンド（[-b]オプション）にも同じサイズを指定してください。
- 固定長ブロックサイズは、512byteの倍数で指定します。ブロックサイズは、使用するテープ装置ごとに最大値が異なります。詳細は各テープ装置のマニュアルをお読みください。

1. テープが書込み可能な状態になっていることを確認し、テープ装置に挿入します。
   RHELシステムがテープデバイスを認識しない場合、次のコマンドを実行し、デバイスファイルを作成します。

```
# mknod /dev/nst0 c 9 128
```

   cはキャラクターデバイス、9はテープ装置、128は装置番号を意味します。

2. テープを巻き戻し、バックアップデータの書込み位置をテープの先頭に位置づけます。

```
# mt -f /dev/nst0 rewind
```

3. テープユニットの状態を表示し、ファイル番号（File number）を確認します。

```
# mt -f /dev/nst0 status
```

テープの先頭からバックアップデータを格納するため、最初のバックアップはファイル番号（File number）が0であることを確認します。

4. ディスクパーティションをバックアップします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# dump 0uf /dev/nst0 ＜パーティションの互換デバイス名＞
```

例：vfatファイルシステムの場合

```
# cd /boot/efi
# tar czvf /dev/nst0 .
```

- dumpコマンドまたはtarコマンドにブロックサイズを指定する場合は、[-b]オプションを使用してください。

5. テープユニットの状態を表示し、ファイル番号（File number）を確認します。

このファイル番号が、次のバックアップデータの書込み位置になります。なお、statusオプションのmtコマンドを実行しても、テープの書込み位置に変化はありません。

```
# mt -f /dev/nst0 status
```

dumpコマンド実行前と比較して、ファイル番号が1加算されたことを確認します。

- 互換デバイス名/dev/st0を使った場合、コマンドが終了したときに、テープが巻き戻されます。このときは、dumpコマンド実行後のファイル番号に戻して、次のバックアップデータの書込みの準備をします。書込み位置を変更したあとは、テープユニットの情報を確認します。

  ```
  # mt -f /dev/nst0 asf ＜ファイル番号＞
  ```

  ＜ファイル番号＞は、dumpコマンド実行前のファイル番号に1加算した値です。

- ファイル番号とバックアップしたディスクパーティションの対応は、リストアのときに必要ですので、必ず記録してください。

6. すべてのディスクパーティションをバックアップするまで、手順4から手順5を繰り返します。

なお、バックアップの途中でテープを使い切った場合、新しいテープに入れ替えて、バックアップを継続します。

7. すべてのディスクパーティションのバックアップが完了したら、テープを取り出します。

```
# mt -f /dev/nst0 offline
```

バックアップに使用したテープは、ファイル番号とディスクパーティションの対応情報と共に、書込み禁止状態で保管します。

8. RHELシステムを再起動して、通常運用状態に戻します。

```
# shutdown -r now
```

7.3.2 テープ装置を使用したリモートバックアップ

リモート接続のテープ装置（DAT、LTO）を使用して、ディスクパーティションをバックアップする手順を説明します。

- バックアップとリストアをそれぞれ異なるテープ装置で実行する場合、テープ装置のデフォルトのブロックサイズが異なる場合があります。バックアップ時のブロックサイズとリストア時のブロックサイズが異なると、リストアできませんので、バックアップとリストアのブロックサイズを一致させてください。
- テープ装置が他のサーバと共有されているなどの理由で、他者によってテープ装置のブロックサイズが変更されていることがあります。その場合は、テープ装置が他のサーバで利用されていない状態で、ブロックサイズを設定してください。

  例：可変長に設定する場合

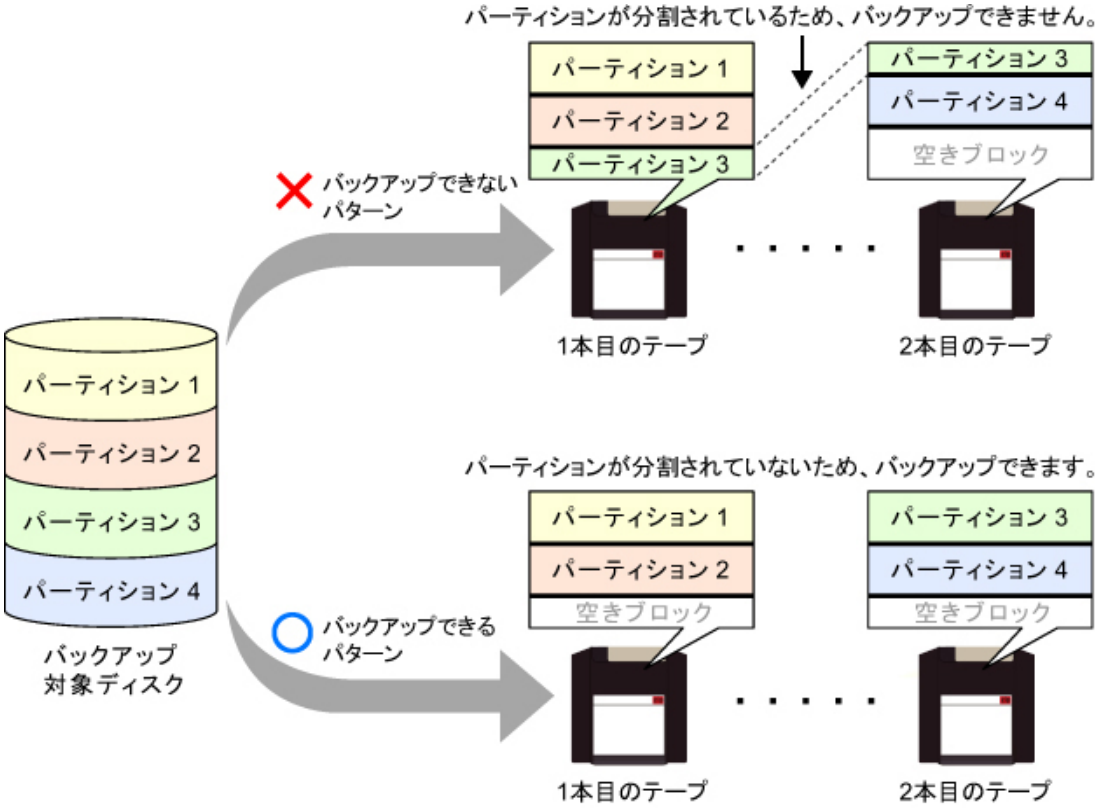
  ```
  # ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 setblk 0
  ```

  例：固定長10240byteに設定する場合

  ```
  # ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 setblk 10240
  ```

- テープ装置が他のサーバと共有されているなどの理由でテープ装置がロックされ、使用できない場合があります。 このときは、ロックされた原因を特定し、使用可能な状態にしてください。
- テープ装置のブロックサイズを固定長に設定する場合は、バックアップに使用するコマンドのブロックサイズにも同じサイズを指定してください。ddコマンドは[obs]オプションでブロックサイズを指定します。
- 固定長ブロックサイズは、512byteの倍数で指定します。ブロックサイズは、使用するテープ装置ごとに最大値が異なります。詳細は各テープ装置のマニュアルをお読みください。
- ネットワークを使用可能な状態にしてください。ネットワークは、セキュリティを維持した環境および業務に支障がないようバックアップ専用LANを用意することを推奨します。
- バックアップサーバへの操作はsshコマンドを使用します。パスワードの問合せがある場合は、パスワードを入力してください。
- ディスクパーティションのバックアップは、コマンド実行中のテープ交換に対応していません。そのため、ディスクパーティションは、必ず1本のテープに収まるようにバックアップしてください。

図 **39 :** テープ装置を使用したリモートバックアップ



1. ネットワークを利用可能にします。

```
# /etc/rc.d/init.d/network start
```

参 考

- レスキューモードの場合、本手順は不要です。

2. テープが書込み可能な状態になっていることを確認し、テープ装置に挿入します。

RHELシステムがテープデバイスを認識しない場合は、次のコマンドを実行し、デバイスファイルを作成します。

```
# ssh ＜IPアドレスまたはホスト名＞ mknod /dev/nst0 c 9 128
```

cはキャラクターデバイス、9はテープ装置、128は装置番号を意味します。

3. テープを巻き戻し、バックアップデータの書込み位置をテープの先頭に位置づけます。

```
# ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 rewind
```

4. テープユニットの状態を表示し、ファイル番号（File number）を確認します。

```
# ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 status
```

テープの先頭からバックアップデータを格納するため、最初のバックアップはファイル番号（File number）が0であることを確認します。

5. ディスクパーティションをバックアップします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# dump 0uf - ＜パーティションの互換デバイス名＞ | ssh ＜IPアドレスまたはホスト名＞ dd
 of=/dev/nst0
```

例：vfatファイルシステムの場合

```
# cd /boot/efi
# tar czvf - . | ssh ＜IPアドレスまたはホスト名＞ dd of=/dev/nst0
```

- dumpコマンドを実行する前に、I/Oスケジューラをdeadlineに変更することで、dumpコマンドの実行速度の向上が期待できます。

  ```
  # echo 'deadline' > /sys/block/sda/queue/scheduler
  ```

  バックアップ対象のデバイスに合わせて、下線部のファイル名を変更します。なお、本設定は一時的な設定であり、RHELシステムの再起動によって破棄されます。

- ddコマンドにブロックサイズを指定する場合は、[obs]オプションを使用してください。

6. テープユニットの状態を表示し、ファイル番号（File number）を確認します。

このファイル番号が、次のバックアップデータの書込み位置になります。なお、status オプションのmtコマンドを実行しても、テープの書込み位置に変化はありません。

```
# ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 status
```

dumpコマンド実行前と比較して、ファイル番号が1加算されたことを確認します。

- 互換デバイス名/dev/st0を使った場合、コマンドが終了したときに、テープが巻き戻されます。このときは、dumpコマンド実行後のファイル番号に戻して、次のバックアップデータの書込みの準備をします。書込み位置を変更したあとは、テープユニットの情報を確認します。

  ```
  # ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 asf ＜ファイル番号＞
  ```

  ＜ファイル番号＞は、dumpコマンド実行前のファイル番号に1加算した値です。

- ファイル番号とバックアップしたディスクパーティションの対応は、リストアのときに必要ですので、必ず記録してください。

7. すべてのディスクパーティションをバックアップするまで、手順5から手順6を繰り返します。

8. すべてのディスクパーティションのバックアップが完了したら、テープを取り出します。

```
# ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 offline
```

バックアップに使用したテープは、ファイル番号とディスクパーティションの対応情報と共に、書込禁止状態で保管します。

9. RHELシステムを再起動して、通常運用状態に戻します。

```
# shutdown -r now
```

7.3.3 データカートリッジを使用したローカルバックアップ

ローカル接続のデータカートリッジを使用して、ディスクパーティションをバックアップする手順について説明します。

1. データカートリッジ内のディスクパーティションをマウントします。

   例：/backupディレクトリを作成してデータカートリッジをマウントします。

```
# mkdir /backup
# mount /dev/sdc1 /backup
```

- この例では、データカートリッジの互換デバイス名を/dev/sdcとし、ディスクパーティション1にバックアップしています。データカートリッジの互換デバイス名の確認方法については、データカートリッジの取扱説明書をお読みください。また、データカートリッジのファイルシステムがRHEL用ではない場合、RHEL用のファイルシステムを作成し、ラベル名を付与してからマウントします。

```
# mkfs.ext3 /dev/sdc1
# e2label /dev/sdc1 /backlabel
```

2. ディスクパーティションをバックアップします。

   例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# dump 0uf /backup/＜バックアップファイル名＞ ＜パーティションの互換デバイス名＞
```

   例：vfatファイルシステムの場合

```
# cd /boot/efi
# tar czvf /backup/backupdata1.tar.gz .
```

3. すべてのディスクパーティションをバックアップするまで、手順2を繰り返します。

4. すべてのディスクパーティションのバックアップが完了したら、データカートリッジをアンマウントして、マウントポイントのディレクトリを削除します。

```
# umount /backup
# rm -rf /backup
```

5. RHELシステムを再起動して、通常運用状態に戻します。

```
# shutdown -r now
```

- 複数のデータカートリッジを使用する場合は、データカートリッジ交換後、次の手順を繰り返します。

「7.2.1 ランレベル1（シングルユーザーモード）への移行」（→P.125）
「7.3.3 データカートリッジを使用したローカルバックアップ」（→P.133）

- データカートリッジは、ディスクをカートリッジ化した媒体ですので、テープ媒体のように、媒体を交換することによる連続的なデータ書込みができません。そのため、データカートリッジには、バックアップするディスクパーティションの使用量よりも大きな空き領域があることを確認してください。
- データカートリッジには、あらかじめラベル名を付与してください。
- データカートリッジを使用できるかどうかの確認は、『システム構成図』をお読みください。

7.3.4 データカートリッジを使用したリモートバックアップ

リモート接続のデータカートリッジを使用して、ディスクパーティションをバックアップする手順を説明します。

1. ネットワークを利用可能にします。

```
# /etc/rc.d/init.d/network start
```

- レスキューモードの場合、本手順は不要です。

2. データカートリッジ内のディスクパーティションをマウントします。

例：/backupディレクトリを作成してデータカートリッジをマウントします。

```
# ssh ＜IPアドレスまたはホスト名＞ mkdir /backup
# ssh ＜IPアドレスまたはホスト名＞ mount /dev/sdc1 /backup
```

- ここでは、データカートリッジの互換デバイス名を/dev/sdcとし、ディスクパーティション1にバックアップしています。データカートリッジの互換デバイス名の確認方法については、データカートリッジの取扱説明書をお読みください。また、データカートリッジのファイルシステムがRHEL用ではない場合、RHEL用のファイルシステムを作成し、ラベル名を付与してからマウントします。

  ```
  # ssh ＜IPアドレスまたはホスト名＞ mkfs.ext3 /dev/sdc1
  # ssh ＜IPアドレスまたはホスト名＞ e2label /dev/sdc1 /backlabel
  ```

3. ディスクパーティションをバックアップします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# dump 0f - ＜パーティションの互換デバイス名＞ | ssh ＜IPアドレスまたはホスト名＞ ¥
dd of=/backup/＜バックアップファイル名＞
```

例：vfatファイルシステムの場合

```
# cd /boot/efi
# tar czvf - . | ssh ＜IPアドレスまたはホスト名＞ dd of=/backup/
backupdata1.tar.gz
```

参考

- dumpコマンドを実行する前に、I/Oスケジューラをdeadlineに変更することで、dumpコマンドの実行速度の向上が期待できます。

```
# echo 'deadline' > /sys/block/sda/queue/scheduler
```

バックアップ対象のデバイスに合わせて、下線部のファイル名を変更します。なお、本設定は一時的な設定であり、RHELシステムの再起動によって破棄されます。

4. すべてのファイルシステムをバックアップするまで、手順3を繰り返します。

5. すべてのディスクパーティションのバックアップが完了したら、データカートリッジをアンマウントして、マウントポイントのディレクトリを削除します。

```
# ssh ＜IPアドレスまたはホスト名＞ umount /backup
# ssh ＜IPアドレスまたはホスト名＞ rm -rf /backup
```

6. RHELシステムを再起動して、通常運用状態に戻します。

```
# shutdown -r now
```

注意

- データカートリッジは、ディスクをカートリッジ化した媒体ですので、テープ媒体のように、媒体を交換することによる連続的なデータ書込みができません。そのため、データカートリッジには、バックアップするディスクパーティションの使用量よりも大きな空き領域があることを確認してください。
- データカートリッジには、あらかじめラベル名を付与してください。
- データカートリッジを使用できるかどうかの確認は、『システム構成図』をお読みください。
- ネットワークを使用可能な状態にしてください。ネットワークは、セキュリティを維持した環境および業務に支障がないようバックアップ専用LANを用意することを推奨します。
- バックアップサーバへの操作はsshコマンドを使用します。パスワードの問合せがある場合は、パスワードを入力してください。

7.4 リストア前の準備

バックアップデータをリストアする前に必要な操作について説明します。

7.4.1 システムの起動

リストアする場合は、リストア対象によって起動方法が変わります。リストア対象に応じて起動方法を適宜選択し、実施します。

■ システムディスクパーティションをリストアする場合

システムディスクパーティションをリストアする場合、RHELシステムをレスキューモードで起動します。 レスキューモードの起動方法については、「6.5 レスキューモードの起動（トラブル発生時）」（→P.117）をお読みください。

■ システムディスクパーティション以外をリストアする場合

システムディスクパーティション以外をリストアする場合は、RHELシステムをランレベル1状態で起動します。移行方法については、「7.2.1 ランレベル1（シングルユーザーモード）への移行」（→P.125）をお読みください。

- システムをランレベル1（シングルユーザーモード）に移行するために、initコマンドは使用しないでください。initコマンドを使用した場合、サービスが完全に停止しない可能性があります。また、安全のため、ランレベル1へ移行したあとには、リストア対象となる資源が使用中になっていないことを必ず確認してください。

7.4.2 ディスク交換後の注意事項

ディスク障害などによりディスクを交換した場合、ディスクパーティション作成時の注意事項について説明します。

ディスク交換前と同じディスクパーティション環境を構築するためには、「7.1.4 バックアップ前の確認事項」（→P.123）で確認したディスクパーティションの詳細情報を使用します。この情報を基に、partedコマンドまたはfdiskコマンドを使用して、ディスクパーティションを作成します。ディスクパーティションの作成方法については、「1.1 ディスクパーティション」（→P.13）をお読みください。

■ パーティションテーブルがMS-DOS形式の場合

ファイルシステムの作成は、fdiskコマンドを使用します。fdiskコマンドを使用する場合の注意事項は、次のとおりです。

- レスキューモードで実行した場合、新しいディスクパーティション情報を反映するため、パーティション分割の終了後、RHELシステムを再起動します。その後、再度レスキューモードで起動します。
- システムディスクパーティションの場合は、ディスクラベルをmsdosにしてください。システムパーティション以外の場合は、ディスクラベルの設定は任意です。なお、ディスクラベルの設定は、partedコマンドを使用してください。

■ パーティションテーブルがGPT形式の場合

ファイルシステムの作成は、partedコマンドを使用します。partedコマンドを使用する場合の注意事項は、次のとおりです。

- レスキューモードで実行した場合、新しいディスクパーティション情報を反映するため、パーティション分割の終了後、RHELシステムを再起動し、改めてレスキューモードで起動してください。
- システムディスクパーティションの場合、ディスクラベルをgptにしてください。システムパーティション以外の場合は、ディスクラベルの設定は任意です。

7.4.3 リストアするディスクパーティションのファイルシステム作成

バックアップデータをリストアする前に、ディスクパーティションにファイルシステムを作成します。これにより、ディスク交換を伴わない場合は、既存のすべてのデータを削除することになり、ディスク交換の場合はファイルシステムを新たに作成することになります。

ファイルシステムの詳細情報は、「7.1.4 バックアップ前の確認事項」（→P.123）で確認した詳細情報を使用します。

■ vfatファイルシステムの作成

vfatファイルシステムの作成手順を説明します。

1. vfatファイルシステムを作成します。

```
# mkfs.vfat /dev/sda1
```

2. ファイルシステムのボリュームIDを設定します。

```
# mkdosfs -i ＜ボリュームID＞ /dev/sda1
```

- ボリュームIDは、blkidコマンドで出力されるボリュームIDから「-（ハイフン）」を削除した値を設定してください。

3. ファイルシステムのラベル名を設定します。

```
# mkdosfs -n /boot/efi /dev/sda1
```

■ ext3ファイルシステムの作成

ext3ファイルシステムの作成手順を説明します。

1. ext3ファイルシステムを作成します。

```
# mkfs.ext3 /dev/sda1
# mkfs.ext3 /dev/sda2
```

- ext2ファイルシステムを作成する場合は、mkfs.ext2コマンドを使用します。ext4ファイルシステムを作成する場合は、mkfs.ext4コマンドを使用します。

2. ファイルシステムのUUIDを設定します。

```
# tune2fs -U ＜/dev/sda1のUUID＞ /dev/sda1
# tune2fs -U ＜/dev/sda2のUUID＞ /dev/sda2
```

3. ファイルシステムのラベル名を設定します。

```
# e2label /dev/sda1 /boot
# e2label /dev/sda2 /
```

■ swapパーティション

swapパーティションの作成手順を説明します。

- バックアップしたswapパーティションにUUIDが設定されていた場合
  [-U]オプションでUUIDを指定します。

```
# mkswap -U ＜swapのUUID＞ /dev/sda3
```

- バックアップしたswapパーティションにUUIDとラベル名が設定されていた場合

[-U]オプションでUUIDを指定し、[-L]オプションでラベル名を指定します。

```
# mkswap -U ＜swapのUUID＞ -L SWAP-sda3 /dev/sda3
```

7.5 リストアの実施

ここでは、リストアの手順について説明します。
リストアは、ローカルリストア、リモートリストア、およびリストア媒体の種類（テープ装置またはデータカートリッジ）によって手順が異なります。環境に応じたバックアップ手順を確認してください。

- mtコマンドおよびrestoreコマンドに指定するテープ装置（DAT、LTO）の互換デバイス名は、コマンド終了時のテープ巻戻しの扱いにより、次の2種類を使い分けます。
  - /dev/st0
    コマンド終了時に、テープを巻き戻す場合に使用します。
  - /dev/nst0
    コマンド終了時に、テープを巻き戻さずに、そのままにする場合に使用します。

7.5.1 テープ装置を使用したローカルリストア

ローカル接続のテープ装置（DAT、LTO）を使用して、ディスクパーティションをリストアする手順を説明します。

- バックアップとリストアをそれぞれ異なるテープ装置で実行する場合、テープ装置のデフォルトのブロックサイズが異なる場合があります。バックアップ時のブロックサイズとリストア時のブロックサイズが異なると、リストアできませんので、バックアップとリストアのブロックサイズを一致させてください。
- テープ装置が他のサーバと共有されているなどの理由で、他者によってテープ装置のブロックサイズが変更されていることがあります。その場合は、テープ装置が他のサーバで利用されていない状態で、ブロックサイズを設定してください。
  例：可変長に設定する場合

  ```
  # mt -f /tmp/nst0 setblk 0
  ```

  例：固定長10240byteに設定する場合

  ```
  # mt -f /tmp/nst0 setblk 10240
  ```
- テープ装置が他のサーバと共有されているなどの理由でテープ装置がロックされ、使用できない場合があります。 このときは、ロックされた原因を特定し、使用可能な状態にしてください。
- バックアップ時に実行したコマンドでブロックサイズを指定した場合は、リストア時にも同じブロックサイズを指定してください。restoreコマンドおよびtarコマンドの場合、[-b]オプションを使用します。

1. テープが書込み禁止状態であることを確認し、リストアするディスクパーティションのバックアップデータを含むテープをテープ装置に挿入します。

2. レスキューモードの場合、テープデバイスを認識しない場合があります。その場合は、次のコマンドを実行してデバイスファイルを作成します。

これ以降、/tmp/nst0ファイルを使用して説明します。ランレベル1でリストアする場合、/dev/nst0ファイルと読み替えてください。

```
# mknod /tmp/nst0 c 9 128
```

cはキャラクターデバイス、9はテープ装置、128は装置番号を意味します。

レスキューモードの場合、RHELのインストール媒体から起動するので、ディスクから起動した場合と異なり、/devディレクトリが存在しない場合があります。そのため、必ず存在する/tmpディレクトリを使用して、デバイスファイルを作成します。

3. テープを巻き戻し、バックアップデータの読込み位置をテープの先頭に位置づけます。

```
# mt -f /tmp/nst0 rewind
```

4. テープの読込み位置に位置づけ、ファイル番号（File number）を確認します。

```
# mt -f /tmp/nst0 asf ＜ファイル番号＞
# mt -f /tmp/nst0 status
```

「7.3.1 テープ装置を使用したローカルバックアップ」（→P.128）の記録から、リストアするディスクパーティションのバックアップデータを格納したファイル番号（File number）を確認します。

5. リストアするディスクパーティションのマウント用ディレクトリを作成します。

```
# mkdir /mnt/work
```

6. リストアするディスクパーティションをマウントします。

例：ext3ファイルシステムの場合

```
# mount -t ext3 ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

例：ext4ファイルシステムの場合

```
# mount -t ext4 ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

例：vfatファイルシステムの場合

```
# mount -t vfat ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

7. ファイルシステムをリストアします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# cd /mnt/work
# restore rf /tmp/nst0
```

restoreコマンド実行後、「restore:/lost+found:File exists」のメッセージが出力される場合がありますが、問題ありませんので、無視します。

例：vfatファイルシステムの場合

```
# cd /mnt/work
# tar xzvf /tmp/nst0
```

- restoreコマンドまたはtarコマンドにブロックサイズを指定する場合は、[-b]オプションを使用してください。
- restoreコマンドを実行すると、カレントディレクトリ配下にrestoresymtableファイルが作成されます。このファイルは、すべてのリストア作業が終了するまで削除しないでください。
- /tmpファイルシステムが独立したパーティションの場合、リストアした際にパーミッションが変更される可能性があります。パーミッションが変更されると、/tmpディレクトリを使用するサービスが起動できなくなるなどの影響があります。そのため、リストアしたあとに、/tmpディレクトリのパーミッションを1777に戻してください。/tmpディレクトリのパーミッション変更手順は、次のとおりです。

```
# cd /
# ls -l | grep tmp
dr--r--r-- 3 root root 4096 11月 16 00:00 2010 tmp
# chmod 1777 /tmp
# ls -l | grep tmp
drwxrwxrwt 3 root root 4096 11月 16 00:00 2010 tmp
```

- restoreコマンドは、/tmpファイルシステムの容量が不足した場合、次のエラーメッセージを表示します。この場合、使用していないパーティションまたは新規パーティションを使用することで、一時的に/tmp領域を拡張してください。なお、テープデバイスを手順2で作成している場合、/tmp/nst0ファイルを削除してから/tmpファイルシステムを拡張します。/tmpファイルシステムの拡張後は、手順2から再開してください。

```
restore: cannot write to file /tmp/＜任意のファイル名＞: no space left on
 device
```

8. リストアしたファイルシステムの整合性をチェックします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# cd /
# umount /mnt/work
# e2fsck -nfv ＜リストアしたディスクパーティションの互換デバイス名＞
# e2fsck -pfv ＜リストアしたディスクパーティションの互換デバイス名＞
```

例：vfatファイルシステムの場合

```
# cd /
# umount /mnt/work
# fsck -pfv ＜リストアしたディスクパーティションの互換デバイス名＞
```

- e2fsckコマンドは、ファイルシステムのチェック（[-n]オプション）、データ削除を伴わない修復（[-p]オプション）の順序で実行します。
- e2fsckコマンドの[-y]オプションの使用は推奨しません。[-y]オプションは、データ削除を伴う修復も実施するので、使用者が意図しないデータ削除の可能性があります。

9. すべてのリストアが終了するまで、手順1から手順8または手順4から手順8を繰り返します。

複数のテープに渡るバックアップデータの場合は、画面の指示に従ってテープを入れ替えます。

10. すべてのリストアが終了したら、テープを取り出します。

```
# mt -f /tmp/nst0 offline
# rm -rf /tmp/nst0
```

11. 一時的に作成したファイルを削除します。

```
# cd /mnt/work
# rm restoresymtable
# rm -rf /tmp/nst0
# rm -rf /mnt/work
```

12. RHELシステムを再起動します。

```
# reboot
```

レスキューモードの場合、ディスクパーティションからRHELを起動するために、RHELのインストール媒体を取り出します。

13. システムディスクパーティションをリストアした場合は、「7.5.5 システムディスクパーティションリストア後の操作」（→P.149）を実施します。

7.5.2 テープ装置を使用したリモートリストア

リモート接続のテープ装置（DAT、LTO）を使用して、ディスクパーティションをリストアする手順を説明します。

注 意

- バックアップとリストアをそれぞれ異なるテープ装置で実行する場合、テープ装置のデフォルトのブロックサイズが異なる場合があります。バックアップ時のブロックサイズとリストア時のブロックサイズが異なると、リストアできませんので、バックアップとリストアのブロックサイズを一致させてください。
- テープ装置が他のサーバと共有されているなどの理由で、他者によってテープ装置のブロックサイズが変更されていることがあります。その場合は、テープ装置が他のサーバで利用されていない状態で、ブロックサイズを設定してください。

  例：可変長に設定する場合

  ```
  # ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 setblk 0
  ```

  例：固定長10240byteに設定する場合

  ```
  # ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 setblk 10240
  ```

- テープ装置が他のサーバと共有されているなどの理由でテープ装置がロックされ、使用できない場合があります。 このときは、ロックされた原因を特定し、使用可能な状態にしてください。
- バックアップ時に実行したコマンドでブロックサイズを指定した場合は、リストア時にも同じブロックサイズを指定してください。ddコマンドの場合、[ibs]オプションを使用します。
- ネットワークを使用可能な状態にしてください。ネットワークは、セキュリティを維持した環境および業務に支障がないようバックアップ専用LANを用意することを推奨します。
- バックアップサーバへの操作はsshコマンドを使用します。パスワードの問合せがある場合は、パスワードを入力してください。

1. ネットワークを使用可能にします。

```
# /etc/rc.d/init.d/network start
```

- レスキューモードの場合、本手順は不要です。

2. テープが書込み可能な状態になっていることを確認し、テープ装置に挿入します。

RHELシステムがテープデバイスを認識しない場合は、次のコマンドを実行し、デバイスファイルを作成します。

```
# ssh ＜IPアドレスまたはホスト名＞ mknod /dev/nst0 c 9 128
```

cはキャラクターデバイス、9はテープ装置、128は装置番号を意味します。

3. テープを巻き戻し、バックアップデータの読込み位置をテープの先頭に位置づけます。

```
# ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 rewind
```

4. テープの読込み位置に位置づけ、ファイル番号（File number）を確認します。

```
# ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 asf ＜ファイル番号＞
# ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 status
```

「7.3.2 テープ装置を使用したリモートバックアップ」（→P.130）の記録から、リストアするディスクパーティションのバックアップデータを格納したファイル番号（File number）を確認します。

5. リストアするディスクパーティションのマウント用ディレクトリを作成します。

```
# mkdir /mnt/work
```

6. リストアするディスクパーティションをマウントします。

例：ext3ファイルシステムの場合

```
# mount -t ext3 ＜リストアするパーティションデバイス名＞ /mnt/work
```

例：ext4ファイルシステムの場合

```
# mount -t ext4 ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

例：vfatファイルシステムの場合

```
# mount -t vfat ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

7. ファイルシステムをリストアします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# cd /mnt/work
# ssh ＜IPアドレスまたはホスト名＞ dd if=/dev/nst0 | restore rf -
```

restoreコマンド実行後、「restore:/lost+found:File exists」のメッセージが出力される場合がありますが、問題ありませんので、無視します。

例：vfatファイルシステムの場合

```
# cd /mnt/work
# ssh ＜IPアドレスまたはホスト名＞ dd if=/dev/nst0 | tar xzvf -
```

注意

- ddコマンドにブロックサイズを指定する場合は、[ibs]オプションを使用してください。
- restoreコマンドを実行すると、カレントディレクトリ配下にrestoresymtableファイルが作成されます。このファイルは、すべてのリストア作業が終了するまで削除しないでください。
- /tmpファイルシステムが独立したパーティションの場合、リストアした際にパーミッションが変更される可能性があります。パーミッションが変更されると、/tmpディレクトリを使用するサービスが起動できなくなるなどの影響があります。そのため、リストアしたあとに、/tmpディレクトリのパーミッションを1777に戻してください。/tmpディレクトリのパーミッション変更手順は、次のとおりです。

```
# ls -l /mnt/work | grep tmp
dr--r--r-- 109 root root 8192 11月 19 04:02 tmp
# chmod 1777 /mnt/work/tmp
drwxrwxrwt 109 root root 8192 11月 19 04:02 tmp
```

- restoreコマンドは、/tmpファイルシステムが不足した場合、次のエラーメッセージを表示します。この場合、使用していないパーティションまたは新規パーティションを使用することで、一時的に/tmpファイルシステムを拡張してください。/tmpファイルシステムの拡張後は、手順3（テープ交換した場合は、手順2）から再開してください。

```
restore: cannot write to file /tmp/＜任意のファイル名＞: no space left on
 device
```

8. リストアしたファイルシステムの整合性をチェックします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# cd /
# umount /mnt/work
# e2fsck -nfv ＜リストアしたディスクパーティションの互換デバイス名＞
# e2fsck -pfv ＜リストアしたディスクパーティションの互換デバイス名＞
```

例：vfatファイルシステムの場合

```
# cd /
# umount /mnt/work
# fsck -pfv ＜リストアしたディスクパーティションの互換デバイス名＞
```

参考

- e2fsckコマンドは、ファイルシステムのチェック（[-n]オプション）、データ削除を伴わない修復（[-p]オプション）の順序で実行します。
- e2fsckコマンドの[-y]オプションの使用は推奨しません。[-y]オプションは、データ削除を伴う修復も実施するので、使用者が意図しないデータ削除の可能性があります。

9. すべてのリストアが終了するまで、手順2から手順8または手順4から手順8を繰り返します。

10. すべてのリストアが終了したら、テープを取り出します。

```
# ssh ＜IPアドレスまたはホスト名＞ mt -f /dev/nst0 offline
```

11. 一時的に作成したファイルを削除します。

```
# cd /mnt/work
# rm restoresymtable
# rm -rf /mnt/work
```

12. RHELシステムを再起動します。

```
# reboot
```

レスキューモードの場合、ディスクパーティションからRHELを起動するために、RHELのインストール媒体を取り出します。

13. システムディスクパーティションをリストアした場合は、続けて「7.5.5 システムディスクパーティションリストア後の操作」（→P.149）を実施します。

7.5.3 データカートリッジを使用したローカルリストア

ローカル接続のデータカートリッジを使用して、ディスクパーティションをリストアする手順を説明します。

1. データカートリッジ内のディスクパーティションをマウントします。

例：/backupディレクトリを作成してデータカートリッジをマウントします。

```
# mkdir /backup
# mount /dev/sdc1 /backup
```

- ここでは、データカートリッジの互換デバイスを /dev/sdc とし、ディスクパーティション 1からリストアしています。データカートリッジの互換デバイス名の確認方法については、データカートリッジの取扱説明書をお読みください。

2. データカートリッジのラベル名を確認します。

ラベル名を使い、手順1でマウントしたデータカートリッジ内のディスクパーティションが、バックアップデータ保存用データカートリッジであることを確認します。

```
# e2label /dev/sdc1
/backlabel
```

3. リストアするディスクパーティションのマウント用ディレクトリを作成します。

```
# mkdir /mnt/work
```

4. リストアするディスクパーティションをマウントします。

例：ext3ファイルシステムの場合

```
# mount -t ext3 ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

例：ext4ファイルシステムの場合

```
# mount -t ext4 ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

例：vfatファイルシステムの場合

```
# mount -t vfat ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

5. ファイルシステムをリストアします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# cd /mnt/work
# restore rf /backup/＜バックアップファイル名＞
```

restoreコマンド実行後、「restore:/lost+found:File exists」のメッセージが出力される場合がありますが、問題ありませんので、無視します。

例：vfatファイルシステムの場合

```
# cd /mnt/work
# tar xzvf /backup/＜バックアップファイル名＞
```

- restoreコマンドを実行すると、カレントディレクトリ配下にrestoresymtableファイルが作成されます。このファイルは、すべてのリストア作業が終了するまで削除しないでください。
- /tmpファイルシステムが独立したパーティションの場合、リストアした際にパーミッションが変更される可能性があります。パーミッションが変更されると、/tmpディレクトリを使用するサービスが起動できなくなるなどの影響があります。そのため、リストアしたあとに、/tmpディレクトリのパーミッションを1777に戻してください。/tmpディレクトリのパーミッション変更手順は、次のとおりです。

  ```
  # cd /
  # ls -l | grep tmp
  dr--r--r-- 3 root root 4096 11月 16 00:00 2010 tmp
  # chmod 1777 /tmp
  # ls -l | grep tmp
  drwxrwxrwt 3 root root 4096 11月 16 00:00 2010 tmp
  ```

- restoreコマンドは、/tmpファイルシステムの容量が不足した場合、次のエラーメッセージを表示します。この場合、使用していないパーティションまたは新規パーティションを使用することで、一時的に/tmpファイルシステムの容量を拡張してください。/tmpファイルシステムの容量拡張後、restoreコマンドを再実行してください。

  ```
  restore: cannot write to file /tmp/＜任意のファイル名＞: no space left on
   device
  ```

6. リストアしたファイルシステムの整合性をチェックします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# cd /
# umount /mnt/work
# e2fsck -nfv ＜リストアしたディスクパーティションの互換デバイス名＞
# e2fsck -pfv ＜リストアしたディスクパーティションの互換デバイス名＞
```

例：vfatファイルシステムの場合

```
# cd /
# umount /mnt/work
# fsck -pfv ＜リストアしたディスクパーティションの互換デバイス名＞
```

- e2fsckコマンドは、ファイルシステムのチェック（[-n]オプション）、データ削除を伴わない修復（[-p]オプション）の順序で実行します。

- e2fsckコマンドの[-y]オプションの使用は推奨しません。[-y]オプションは、データ削除を伴う修復も実施するので、使用者が意図しないデータ削除の可能性があります。

7. すべてのリストアが終了するまで、手順4から手順6を繰り返します。

8. すべてのリストアが終了したら、データカートリッジをアンマウントします。

```
# umount /backup
```

9. 一時的に作成したファイルを削除します。

```
# cd /mnt/work
# rm restoresymtable
# rm -rf /mnt/work
# rm -rf /backup
```

10. RHELシステムを再起動します。

```
# reboot
```

レスキューモードの場合、ディスクパーティションからRHELを起動するために、RHELのインストール媒体を取り出します。

11. システムディスクパーティションをリストアした場合は、続けて「7.5.5 システムディスクパーティションリストア後の操作」（→P.149）を実施します。

参考

- 複数のデータカートリッジを使用する場合、データカートリッジを交換するたびにレスキューモードで起動し直したあと、手順1から実行します。

7.5.4 データカートリッジを使用したリモートリストア

リモート接続のデータカートリッジを使用して、ディスクパーティションをリストアする手順について説明します。

1. ネットワークを使用可能にします。

```
# /etc/rc.d/init.d/network start
```

注意

- レスキューモードの場合、本手順は不要です。

2. データカートリッジ内のディスクパーティションをマウントします。

例：/backupディレクトリを作成してデータカートリッジをマウントします。

```
# ssh ＜IPアドレスまたはホスト名＞ mkdir /backup
# ssh ＜IPアドレスまたはホスト名＞ mount /dev/sdc1 /backup
```

- ここでは、データカートリッジの互換デバイスを/dev/sdcとし、ディスクパーティション 1からリストアしています。データカートリッジの互換デバイス名の確認方法については、データカートリッジの取扱説明書をお読みください。

3. データカートリッジのラベル名を確認します。

ラベル名を使い、手順2でマウントしたデータカートリッジ内のディスクパーティションが、バックアップデータ保存用データカートリッジであることを確認します。

```
# ssh ＜IPアドレスまたはホスト名＞ e2label /dev/sdc1 /backlabel
```

4. リストアするディスクパーティションのマウント用ディレクトリを作成します。

```
# mkdir /mnt/work
```

5. リストアするディスクパーティションをマウントします。

例：ext3ファイルシステムの場合

```
# mount -t ext3 ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

例：ext4ファイルシステムの場合

```
# mount -t ext4 ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

例：vfatファイルシステムの場合

```
# mount -t vfat ＜リストアするディスクパーティションの互換デバイス名＞ /mnt/work
```

6. ファイルシステムをリストアします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# cd /mnt/work
# ssh ＜IPアドレスまたはホスト名＞ dd if=/backup/＜バックアップファイル名＞ |restore
 rf -
```

restoreコマンド実行後、「restore:/lost+found:File exists」のメッセージが出力される場合がありますが、問題ありませんので、無視します。

例：vfatファイルシステムの場合

```
# cd /mnt/work
# ssh ＜IPアドレスまたはホスト名＞ dd if=/dev/nst0 | tar xzvf -
```

- restoreコマンドを実行すると、カレントディレクトリ配下にrestoresymtableファイルが作成されます。このファイルは、すべてのリストア作業が終了するまで削除しないでください。
- /tmpファイルシステムが独立したパーティションの場合、リストアした際にパーミッションが変更される可能性があります。パーミッションが変更されると、/tmpディレクトリを使用するサービスが起動できなくなるなどの影響があります。そのため、リストアしたあとに、/tmpディレク

トリのパーミッションを1777に戻してください。/tmpディレクトリのパーミッション変更手順は、次のとおりです。

```
# ls -l /mnt/work | grep tmp
dr--r--r-- 109 root root 8192 11月 19 04:02 tmp
# chmod 1777 /mnt/work/tmp
drwxrwxrwt 109 root root 8192 11月 19 04:02 tmp
```

- restoreコマンドは、/tmpファイルシステムの容量が不足した場合、次のエラーメッセージを表示します。この場合、使用していないパーティションまたは新規パーティションを使用することで、一時的に/tmpファイルシステムの容量を拡張してください。/tmpファイルシステムの容量を拡張後、restoreコマンドを再実行してください。

```
restore: cannot write to file /tmp/＜任意のファイル名＞: no space left on
 device
```

7. リストアしたファイルシステムの整合性をチェックします。

例：ext3ファイルシステムまたはext4ファイルシステムの場合

```
# cd /
# umount /mnt/work
# e2fsck -nfv ＜リストアしたパーティションデバイス名＞
# e2fsck -pfv ＜リストアしたパーティションデバイス名＞
```

例：vfatファイルシステムの場合

```
# cd /
# umount /mnt/work
# fsck -pfv ＜リストアしたディスクパーティションの互換デバイス名＞
```

- e2fsckコマンドは、ファイルシステムのチェック（[-n]オプション）、データ削除を伴わない修復（[-p]オプション）の順序で実行します。
- e2fsckコマンドの[-y]オプションの使用は推奨しません。[-y]オプションは、データ削除を伴う修復も実施するので、使用者が意図しないデータ削除の可能性があります。

8. すべてのリストアが終了するまで、手順5から手順7を繰り返します。

9. すべてのリストアが終了したら、データカートリッジをアンマウントします。

```
# ssh ＜IPアドレスまたはホスト名＞ umount /backup
```

10. 一時的に作成したファイルを削除します。

```
# ssh ＜IPアドレスまたはホスト名＞ cd /mnt/work
# ssh ＜IPアドレスまたはホスト名＞ rm restoresymtable
# ssh ＜IPアドレスまたはホスト名＞ rm -rf /mnt/work
# ssh ＜IPアドレスまたはホスト名＞ rm -rf /backup
```

11. RHELシステムを再起動します。

```
# reboot
```

レスキューモードの場合、ディスクパーティションからRHELを起動するために、RHELのインストール媒体を取り出します。

12. システムディスクパーティションをリストアした場合は、続けて「7.5.5 システムディスクパーティションリストア後の操作」（→P.149）を実施します。

注意

- ネットワークを使用可能な状態にしてください。ネットワークは、セキュリティを維持した環境および業務に支障がないようバックアップ専用LANを用意することを推奨します。
- バックアップサーバへの操作はsshコマンドを使用します。パスワードの問合せがある場合は、パスワードを入力してください。

7.5.5 システムディスクパーティションリストア後の操作

システムディスクパーティションをリストアした場合、UEFIまたはブートローダ(GRUB)を設定します。

■ ファームウェアがUEFIモードの場合

UEFIを設定します。
UEFI の詳細については、『運用管理ツールリファレンス』をお読みください。

■ ファームウェアがBIOS互換モードの場合

ブートローダ(GRUB)の設定方法を説明します。

1. レスキューモードでRHELシステムを起動します。
   レスキューモードの起動方法については、「6.5 レスキューモードの起動（トラブル発生時）」（→P.117）をお読みください。

2. 「Rescue」画面で既存のディスクパーティションを自動的にマウントするかどうかのメッセージが表示されたら、［Continue］を選択します。
   リストアしたファイルシステムの/（ルート）パーティションが/mnt/sysimageディレクトリにマウントできた場合は、結果が表示されるので、［OK］を選択します。

   注意

   - リストア後、/etc/fstabファイルに記述しているデバイスのうち、未接続のデバイスがある場合、マウントエラーが発生する可能性があります。エラーが発生した場合は、当該デバイスを記載している行をコメント行に変更して、手順1から再実行してください。

3. 「First Aid Kit quickstart menu」画面で［shell Start shell］を選択します。
   コマンドプロンプトが表示されます。

4. /mnt/sysimageファイルシステムおよび/mnt/sysimage/bootファイルシステムがマウントされていることを確認します。

```
# df
```

5. ルートパスを/mnt/sysimageディレクトリに変更します。

```
# chroot /mnt/sysimage
```

6. ブートローダをMBRにインストールします。

```
# grub-install /dev/sda
```

/dev/sdaはMBRが存在する互換デバイス名です。

7. 手順2で/etc/fstabファイルを編集した場合は元に戻します。

8. chroot環境を終了します。

```
# exit
```

9. RHELシステムを再起動します。

```
# reboot
```

- 手順2で/etc/fstabファイルを編集した場合は、RHELシステムを停止させて、必要なデバイスを接続してからRHELシステムを起動します。

用語解説

本文中で使用している専門用語について説明します。

システムボリューム
: OSがインストールされるハードディスク領域のうち、OSの稼働に必要な最低限のプログラム・データが格納されている領域。OS稼働中はシステムから切り離すことができない。

修正ユリウス日
: 紀元前4713年1月1日正午（世界標準時による）からの日数であるユリウス通日（AJD）から2400000.5を引いたもの。現在使われているグレゴリオ暦から修正ユリウス日を次の式で求めることができる。xを超えない最大の整数を[x]で表すと、グレゴリオ暦y年m月d日午前0時の修正ユリウス日は、次のとおりである。

  [365.25y]+[y/400]-[y/100]+[30.59(m-2)]+d-678912

  ただし、1月、2月の場合は、前年の13月、14月として計算する。

  参考：2001年1月1日の修正ユリウス日は51910

データボリューム
: 主にユーザーの業務用に作成される、アプリケーションやデータが格納されているハードディスク領域。

トランキング
: trunking

  ネットワークのスイッチなどにおいて、隣接した複数のポートを多重化し、一本の高速リンクとして機能させること。

GDS
: Global Disk Services

  製品名はPRIMECLUSTER GDS。富士通が提供する、シングルシステムでのストレージの冗長化を行うボリューム管理ソフトウェア。

GLS
: Global Link Services

  製品名はPRIMECLUSTER GLS。富士通が提供する、ネットワーク伝送路を冗長化し、通信の高信頼化を実現するソフトウェア。

GPS
: Global Positioning System

  地球を周回する20数個の衛星から発信される電波を利用し、緯度、経度、高度、時刻などを割り出すことができるシステム。セシウムなどを使った高精度な原子時計が装備されている。

GSPB
: Giga-LAN SAS and PCI_Box connector Board

  オンボードI/OおよびPCI_Box用PCI Expressインターフェースを搭載したボード。

masterデバイス
: bondingドライバが構成した仮想的なNICを示す。通常はbond＜n＞という形式のデバイス名。

MMB
: ManageMent Board

筐体内の制御・監視、パーティション管理、システム初期化などを行うシステム制御ユニット。

MSTP

Multiple Spanning Tree Protocol

STPプロトコルをVLANを意識して動作するように拡張したプロトコル。物理的には閉路を形成していても、VLAN構成によってはVLAN単位には閉路にならない場合などがあり、こういったケースでもVLAN単位に閉路を検出して切断するためのプロトコルである。

NIC

Network Interface Card

コンピュータネットワーク内でコンピュータ間の通信を行うために使用されるハードウェアの1つ。

PSA

PRIMEQUEST Server Agent

パーティション上のハードウェア異常監視・構成管理などを行う。ハードウェア監視・構成管理は下位のドライバ層からなる。

SAS

Serial Attached SCSI

SCSI規格のうちの1つであり、コンピュータにハードディスクなどのデバイスを接続するためのインターフェースである。シリアル伝送を行う。

SASU

SASディスクユニット

内蔵ハードディスクドライブとそのコントローラとしてのPCIカードを搭載できるユニット。

slaveデバイス

masterデバイスを構成する個々のNICを示す。通常のeth＜n＞という形式のデバイスがslaveデバイスとして組み込まれる。

STP

Spanning Tree Protocol

スイッチでサポートされるプロトコルの1つ。冗長性のあるスイッチネットワークを構成すると、閉路が形成されることがあり、フレームがループすることがある。STPはスイッチ間でリンクの接続形態と状態を交換して閉路を形成するリンクを切断し、フレームのループを防止するプロトコルである。

UTC

Universal Time, Coordinated

全世界で公式な時刻を意味する。時刻には次の種類がある。

- 世界時（UT1）：天体観測をもとに決められている時刻。
- 国際原子時（TAI）：世界時（UT1）における1958年1月1日0時0分0秒からの経過時間を、高精度なセシウム原子時計でカウントして定めた時刻。
- 協定世界時（UTC）：国際原子時（TAI）に世界時（UT1）とのずれを調整するための「うるう秒」を導入した時刻。現在は、グリニッジ標準時（GMT）に代わって標準になっている。